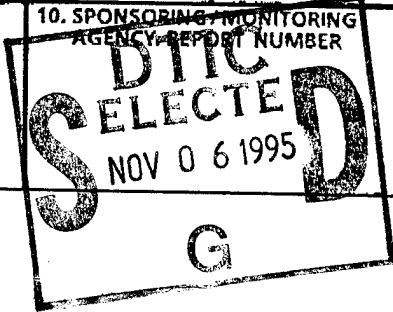


| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 | |
|---|---|--|--|----------------------------------|
| <small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small> | | | | |
| 1. AGENCY USE ONLY (Leave blank) | | 2. REPORT DATE | | 3. REPORT TYPE AND DATES COVERED |
| 4. TITLE AND SUBTITLE Software Reengineering Assessment Method, Version 1.0 | | | 5. FUNDING NUMBERS | |
| 6. AUTHOR(S) Various | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Defense Analyses 1801 N. Beauregard St. Alexandria, VA 22311-1772 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER IDA Paper P-3045 | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Information Systems Agency (DISA) JIEO/Center for Software, Code JEXS 701 South Courthouse Road Arlington, VA 22204-2199 | | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER | |
| 11. SUPPLEMENTARY NOTES | | |  | |
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; Distribution is unlimited. | | | | |
| 13. ABSTRACT (Maximum 200 words) This document describes the System Reengineering Assessment Method (SRAM), which is a method for assessing the costs, risks, and benefits of reengineering information systems. This method explores procedures for managers to assess existing system needs, identify candidate system options, determine comparative costs, risk, and benefits, and finally select a system option that satisfies both system and functional goals. DISA POC: MR BRUCE MORGAN 703-681-2345 (morgan2b@ncr.disa.mil) | | | | |
| 14. SUBJECT TERMS CIM (COLLECTION) | | | 15. NUMBER OF PAGES | |
| | | | 16. PRICE CODE | |
| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT | |

CLEARANCE REQUEST FOR PUBLIC RELEASE OF DEPARTMENT OF DEFENSE INFORMATION

(See Instructions on reverse)

(This form is to be used in requesting review and clearance of DoD information proposed for public release in accordance with DoDD 5230.9.)

TO: Assistant Secretary of Defense (Public Affairs)

ATTN: Director, Freedom of Information & Security Review, Rm. 2C757, Pentagon

DISA

1. DOCUMENT DESCRIPTION

| | |
|----------------------------|---|
| a. TYPE Technical Paper | b. TITLE System Reengineering Assessment Method, Version 1 |
| c. PAGE COUNT 94 | d. SUBJECT AREA Systems Reengineering & Systems Analysis |

2. AUTHOR / SPEAKER

| | | |
|--|----------------------|---|
| a. NAME (Last, First, Middle Initial) Morgan, Bruce, A. | b. RANK GS-334-14 | c. TITLE Computer Specialist |
| d. OFFICE DISA/JIEO/CFSW/JEXSR | | e. AGENCY Defense Information Systems Agency |

3. PRESENTATION / PUBLICATION DATA (Date, Place, Event)

The paper was originally published as a contract deliverable by the Institute for Defense Analysis (IDA) in January 1995. Subsequently, it was revised and republished by the Center for Software in June 1995.

4. POINT OF CONTACT

| | |
|--|--|
| a. NAME (Last, First, Middle Initial) Morgan, Bruce, A. | b. TELEPHONE NO. (Include Area Code) 703-681-2345 |
|--|--|

5. PRIOR COORDINATION

| | | |
|---------------------------------------|--------------------|--------------------------------------|
| a. NAME (Last, First, Middle Initial) | b. OFFICE / AGENCY | c. TELEPHONE NO. (Include Area Code) |
| | | |

6. REMARKS

Request the attached paper be reviewed for open publication and distribution through the Defense Technical Information Center (DTIC) at Cameron Station in Alexandria, VA. This paper is the first version of the Center for Software's System Reengineering Assessment Method.

AS AMENDED - ACCURACY/PP: TITLE PAGE, ES-1, & 1-1.

CLEARED
FOR OPEN PUBLICATION

SEP 07 1995 22

7. RECOMMENDATION OF SUBMITTING OFFICE / AGENCY

a. THE ATTACHED MATERIAL HAS DEPARTMENT / OFFICE / AGENCY APPROVAL FOR PUBLIC RELEASE (qualifications, if any, are indicated in Remarks section) AND CLEARANCE FOR OPEN PUBLICATION IS RECOMMENDED UNDER PROVISIONS OF DoDD 5230.9. I AM AUTHORIZED TO MAKE THIS RECOMMENDATION FOR RELEASE ON BEHALF OF:

DIRECTOR/DISA

b. CLEARANCE IS REQUESTED BY 950818 (YYMMDD).

95-5-3084

| | |
|---|-------------------------------------|
| c. NAME (Last, First, Middle Initial) MCDONALD, BETSY B. | d. TITLE CHIEF OF PUBLIC AFFAIRS |
|---|-------------------------------------|

| | |
|-----------------------------|-------------------|
| e. OFFICE CHIEF OF STAFF | f. AGENCY DISA |
|-----------------------------|-------------------|

g. SIGNATURE

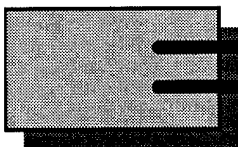
h. DATE SIGNED (YYMMDD)

Janice Bonat for Betsy McDonald 950804

Defense Information Systems Agency
Joint Interoperability & Engineering Organization
Center for Software
701 Courthouse Road; Arlington, VA 22204-2199

System Reengineering Assessment Method

| | |
|--------------------------------------|---|
| Accession For | |
| NTIS | CRA&I <input checked="" type="checkbox"/> |
| DTIC | TAB <input type="checkbox"/> |
| Unannounced <input type="checkbox"/> | |
| Justification | |
| By | |
| Distribution / | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |



Version 1.0



June 1995



UNCLASSIFIED

SYSTEM REENGINEERING ASSESSMENT METHOD

Version 1.0

Defense Information Systems Agency
Joint Interoperability & Engineering Organization
Center for Software

June 1995

"This report is provided for information only.
Substantive contributions will be
incorporated in official instructions and
guidance provided separately."

Prepared for

Defense Information Systems Agency

By

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311

UNCLASSIFIED

PREFACE

This document was prepared by the Institute for Defense Analyses (IDA) under a Center for Software (CFSW) task order, Software Reengineering Methodology. This document fulfills a task objective to develop a method to assess the costs, risks, and benefits of reengineering information systems within the Department of Defense. The work was sponsored by the Defense Information Systems Agency (DISA).

The Center for Software would like to thank the following IDA authors for their contributions: Bill Brykczynski, Kathleen A. Jordan (Task Leader), R.N.C. Lightfoot, Reginald N. Meeson, and Sara H. Nash.

The IDA authors would like to thank Dr. Thomas P. Frazier for his assistance and review of the cost and economic analysis sections of this method. The editorial assistance of Ms. Katydean Price and Ms. Christine Youngblut and the secretarial support of Ms. Edna Jordan are gratefully acknowledged.

The following IDA research staff members were reviewers of this document: Dr. Dennis W. Fife, Ms. Audrey A. Hook, Dr. Richard J. Ivanetich, Dr. Judy M. Popelas, Ms. Beth Springsteen, and Mr. Glen R. White.

The following CFSW staff members reviewed this document: Ms. Cookie Carlson, Ms. Tamra Goldstein, Mr. Bruce Morgan, and Ms. Cynthia Wright.

Table of Contents

| | |
|---|------|
| EXECUTIVE SUMMARY | ES-1 |
| 1. INTRODUCTION | 1-1 |
| 1.1 PURPOSE | 1-1 |
| 1.2 BACKGROUND | 1-1 |
| 1.3 APPROACH | 1-1 |
| 1.4 SCOPE | 1-2 |
| 1.5 AUDIENCE | 1-2 |
| 1.6 NOTATION USED | 1-2 |
| 1.7 ORGANIZATION OF DOCUMENT | 1-3 |
| 2. CONTEXT OF INFORMATION SYSTEM REENGINEERING | 2-1 |
| 2.1 MOTIVATIONS FOR REENGINEERING | 2-1 |
| 2.2 SYSTEM REENGINEERING WITHIN THE INFORMATION MANAGEMENT PROCESS | 2-2 |
| 2.2.1 Information Management Process Model | 2-2 |
| 2.2.2 CIM Software Systems Reengineering Process Model | 2-4 |
| 2.2.3 SRAM Context Diagram | 2-5 |
| 2.3 DEFINITIONS | 2-6 |
| 2.4 RELATED DOCUMENTS | 2-7 |
| 3. SRAM DETAILED DESCRIPTION | 3-1 |
| 3.1 ANALYZE AUTOMATED INFORMATION SYSTEM AND ENVIRONMENT (A1) | 3-4 |
| 3.1.1 Conduct Problem Analysis (A11) | 3-5 |
| 3.1.2 Identify Reengineering Objectives (A12) | 3-8 |
| 3.1.3 Identify Reengineering Constraints (A13) | 3-11 |
| 3.2 IDENTIFY SYSTEM OPTIONS (A2) | 3-12 |
| 3.2.1 Identify System Reengineering Strategies (A21) | 3-13 |
| 3.2.2 Assess Technology Alternatives (A22) | 3-15 |
| 3.2.3 Develop System Configurations (A23) | 3-19 |
| 3.3 ESTIMATE COSTS, RISKS, AND BENEFITS (A3) | 3-21 |
| 3.3.1 Identify and Estimate Risks (A31) | 3-22 |
| 3.3.2 Identify and Estimate Benefits (A32) | 3-25 |
| 3.3.3 Identify and Estimate System Costs(A33) | 3-27 |
| 3.3.4 Identify and Estimate Functional Costs (A34) | 3-33 |
| 3.4 SELECT BEST OPTION(S) (A4) | 3-38 |
| 3.4.1 Determine Decision Criteria (A41) | 3-39 |
| 3.4.2 Rank System Options (A42) | 3-41 |
| 3.4.3 Conduct Sensitivity Analysis (A43) | 3-45 |

| | |
|--|---------|
| 3.4.4 Select Solution/Approach (A44) | 3-47 |
| 4. SUMMARY OF SRAM | 4-1 |
| APPENDIX A. ANALYTICAL HIERARCHY PROCESS EXAMPLE | A-1 |
| LIST OF REFERENCES | Refer-1 |
| GLOSSARY | Gloss-1 |
| LIST OF ACRONYMS | Acro-1 |

List of Figures

| | |
|--|------|
| Figure 1-1. IDEF0 Notation | 1-3 |
| Figure 2-1. Information Management Process Model | 2-2 |
| Figure 2-2. CIM Software Systems Reengineering Process Model | 2-4 |
| Figure 2-3. Context Diagram of SRAM | 2-5 |
| Figure 3-1. Context Diagram of SRAM | 3-1 |
| Figure 3-2. System Reengineering Assessment Method | 3-2 |
| Figure 3-3. Analyze Automated Information System & Environment | 3-4 |
| Figure 3-4. Identify System Options | 3-12 |
| Figure 3-5. Estimate Costs, Risks, and Benefits | 3-21 |
| Figure 3-6. Select Best Option(s) | 3-38 |
| Figure A-1. Reengineering Decision Hierarchy | A-2 |

List of Tables

| | |
|---|------|
| Table 3-1. Application Reengineering Cost Estimation | 3-29 |
| Table 3-2. Data Reengineering Cost Estimation | 3-30 |
| Table 3-3. Infrastructure Reengineering Cost Estimation | 3-31 |
| Table 3-4. FEAM Cost Elements | 3-34 |
| Table A-1. Decision Alternatives | A-3 |
| Table A-2. Major Factors Comparison | A-3 |
| Table A-3. Cost Factors Comparison | A-4 |
| Table A-4. Cost Table | A-4 |
| Table A-5. Pairwise Cost Comparisons | A-5 |
| Table A-6. Operations Cost Matrix | A-5 |
| Table A-7. Maintenance Cost Matrix | A-6 |
| Table A-8. Investment Cost Matrix | A-6 |
| Table A-9. Composite Cost Factors | A-6 |
| Table A-10. Pairwise Benefit Comparisons | A-7 |
| Table A-11. Benefit Matrix | A-7 |
| Table A-12. Pairwise Risk Comparisons | A-8 |
| Table A-13. Risk Matrix | A-8 |
| Table A-14. Cost, Benefit, & Risk Factors | A-9 |
| Table A-15. Decision Factors | A-9 |
| Table A-16. Decision Priorities | A-10 |
| Table A-17. Prioritized Decision List | A-10 |

List of Examples

| | |
|---|------|
| Example 3-1. Problem Statement | 3-6 |
| Example 3-2. Reengineering Objectives | 3-9 |
| Example 3-3. Reengineering Constraints | 3-11 |
| Example 3-4. System Reengineering Strategies | 3-15 |
| Example 3-5. Selected Technologies | 3-19 |
| Example 3-6. Candidate System Options | 3-21 |
| Example 3-7. Risk-Assessed System Options | 3-25 |
| Example 3-8. Benefit-Assessed System Options | 3-27 |
| Example 3-9. Costed System Options | 3-33 |
| Example 3-10. Economic Analysis of System Options | 3-37 |
| Example 3-11. Economic Analysis of System Options (Risk Adjusted) | 3-38 |
| Example 3-12. Selected Decision Criteria | 3-42 |
| Example 3-13. Ranked System Options | 3-46 |
| Example 3-14. Ranked Systems Options - A Portfolio Example | 3-46 |
| Example 3-15. Sensitivity of Rankings. | 3-48 |
| Example 3-16. Selected Option(s) | 3-50 |

EXECUTIVE SUMMARY

Problem

The System Reengineering Assessment Method (SRAM) explores procedures to assess the costs, risks and benefits of reengineering information systems. The SRAM was developed to help modernize the inventory of legacy systems of the Department of Defense (DoD). The number of these systems is overwhelming and the costs of maintaining them have become prohibitive. Most are isolated within single major applications and are non-interoperable with other information systems. Moreover, they were developed before modern methods, languages, and tools were commonly available.

Approach

The Institute for Defense Analyses developed the SRAM by surveying existing software cost models and approaches, reviewing current functional process analysis methods, reviewing candidate reengineering projects, and interviewing managers involved in reengineering efforts. A follow-on phase will refine and validate the method with additional selected reengineering projects.

Information system reengineering builds a "new" system using the existing system as the basis for requirements and design. System reengineering encompasses a combination of other software engineering activities such as reverse engineering, redocumentation, forward engineering, and code translation. At least three general situations may initiate the consideration of reengineering:

- **New Functional Process:** New requirements may have been identified or new work flow patterns may be created, which may require changes to information system support.
- **System Upgrade Required:** A problem with an existing system may have been identified. Examples are the cost of hardware or software maintenance or the non-availability of hardware/software support.
- **Technical Improvement Opportunities:** Technological improvements may suggest potential changes in the functional process. The decision to

reengineer is made within the larger context of the DoD's information management process. This iterative process consists of five steps: (1) perform functional process improvement, (2) perform technical and economic analysis, (3) develop or reengineer system, (4) transition to operation and maintenance, and (5) implement, operate, and maintain.

System Reengineering Assessment Method

The SRAM consists of four major steps:

1. **Analyze Automated System and Environment.** This step is essential to ensuring that the reengineering effort solves the correct problem and considers its scope and limitations before developing possible system solutions. It analyzes all aspects of the problem domain that are to be considered by the reengineering effort. Managers determine reengineering objectives, consider the effect upon the functional process, identify reengineering constraints, and develop a specific problem statement.
2. **Identify System Options.** Managers then identify candidate system options that satisfy the requirements and limitations of the problem. These can consist of a mix of strategies that include new development, continued maintenance, reengineering, and system retirement.
3. **Estimate Costs, Risks, and Benefits.** The costs, risks, and benefits of the candidate system options are then assessed. Cost assessment can be done with a model such as the Functional Economic Analysis Model (FEAM), which assists managers in determining which options produce the best cost savings over a given time period. Risks are assessed with respect to planning, process, product, personnel, and technology. Benefits associated with system functionality, reliability, performance, usability, interoperability, compatibility, and maintainability are also identified and estimated.
4. **Select Best Option(s).** This step consists of four activities: (1) determine decision criteria, (2) rank system options, (3) conduct sensitivity analysis to assess the validity of the rankings, and (4) select

solution/approach, given the system rankings and sensitivity analysis.

The use of the SRAM is illustrated with a system reengineering example. This example, known as the Aircraft Maintenance System (AMS), is provided for each step of the SRAM process.

SRAM Summary

- **Analyze Automated Information System and Environment (A1)**
 - **Conduct Problem Analysis (A11)**
 - **Identify Reengineering Objectives (A12)**
 - **Identify Reengineering Constraints (A13)**
- **Identify System Options (A2)**
 - **Identify System Reengineering Strategies (A21)**
 - **Assess Technology Alternatives (A22)**
 - **Develop System Configurations (A23)**
- **Estimate Costs, Risks, and Benefits (A3)**
 - **Identify and Estimate Risks (A31)**
 - **Identify and Estimate Benefits (A32)**
 - **Identify and Estimate System Costs (A33)**
 - **Identify and Estimate Functional Costs (A34)**
- **Select Best Option(s) (A4)**
 - **Determine Decision Criteria (A41)**
 - **Rank System Options (A42)**
 - **Conduct Sensitivity Analysis (A43)**
 - **Select Solution/Approach (A44)**

1. INTRODUCTION

1.1 PURPOSE

This document describes the System Reengineering Assessment Method (SRAM), which is a method for assessing the costs, risks, and benefits of reengineering information systems. This method explores procedures for managers to assess existing system needs, identify candidate system options, determine comparative costs, risk, and benefits, and finally select a system option that satisfies both system and functional goals.

1.2 BACKGROUND

The development of this method is in response to a number of factors affecting the DoD information management community today. First, there exist within the DoD an overwhelming number of legacy information systems. These systems each tend to be isolated within a single major application and are generally not interoperable with other information systems, leading to a "stovepipe" architecture. Secondly, the costs of maintaining these legacy systems have become prohibitive. These information systems were developed before modern methods, languages, and tools were commonly available. The DoD is currently modernizing its inventory of legacy systems through the Corporate Information Management Initiative, which is seeking to lower systems development, maintenance, and operational costs.

1.3 APPROACH

The SRAM was developed by surveying existing software cost estimation models, reviewing current functional process analysis methods, and reviewing candidate reengineering projects. This effort also interviewed managers involved in reengineering efforts, including the Base-Level System Modernization (BLSM) effort at Standard Systems Center (SSC), Gunter Air Force Base, Alabama and the Defense Logistics Agency (DLA), Cameron Station, Virginia. Additional selected reengineering projects will be used to refine and validate this method in a follow-on phase to this

effort.

1.4 SCOPE

The SRAM describes the decision making process for determining the comparative costs, benefits, and risks for an anticipated system reengineering effort, specifically for those systems within the information systems domain. This method can be used for evaluating situations that apply to both single and multiple systems. Therefore the term "system" within the document can refer to either a single system or multiple systems, as in a "system of systems."

This document is not a guide on the system reengineering process itself, nor a guide on functional (business) process reengineering. However, this method does recognize the role that system reengineering plays within functional process reengineering and discusses the relationship of system reengineering with the functional process. This document is specifically intended to be used with the *Center for Information Management Software Systems Reengineering Process Model, Version 2.0* [CIM94], and with a number of other documents listed in Section 2.4. It should be noted that the SRAM constitutes a set of guidelines, and its use should be adapted to the specific problem at hand.

1.5 AUDIENCE

The audience for this document is the Defense Information Systems Agency and those functional proponents and Central Design Activity (CDA) program managers who are responsible for maintaining DoD information systems. This document is intended for information system professionals who are required to assess, develop, and maintain information systems.

1.6 NOTATION USED

The SRAM process, which is defined in this document, is illustrated with a graphical notation called IDEF0 (see Figure 1-1). This notation depicts the specific activities within a process. Each activity has a set of inputs, controls, outputs, and mechanisms (ICOMs). The inputs are entities that will be either transformed or consumed by the activity. The controls represent entities that are used by the activity, but not transformed by it, such as a regulation or standard. The mechanisms are resources or enablers, such as a database or a person. The outputs are the product of the activity. Although the activity boxes are arranged in a left-to-right

layout, this arrangement does not imply a specific sequence for those activities. The activation of an activity is based upon the availability of inputs and controls. In addition, there can be feedback between activities, where an output of one activity becomes an input, control, or mechanism for an activity that precedes graphically. Also note that ICOMs that are optional are shown with the ICOM labels in parentheses.

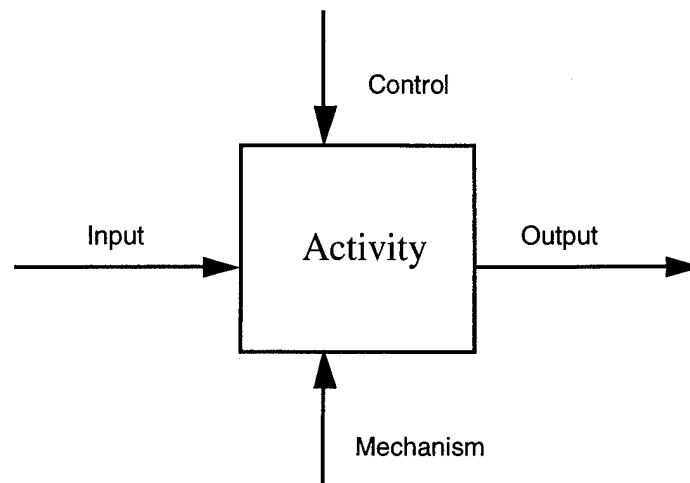


Figure 2-1. IDEF0 Notation

1.7 ORGANIZATION OF DOCUMENT

Section 2 discusses the context of information system reengineering within the information management process, the relationship of functional process improvement to system, and specific definitions for reengineering terms. Section 3 describes the method in detail, providing guidance on developing, assessing, and selecting system options. Appendix A provides an example use of the Analytic Hierarchy Process. References, a glossary, and a list of acronyms conclude the document.

2. CONTEXT OF INFORMATION SYSTEM REENGINEERING

Information system reengineering is essentially building a “new” system using the existing system as the basis for requirements and design. The objective of the SRAM is to determine the specific system option to pursue, based upon relative costs, risks, and benefits. This section discusses the context of using this method, including the motivations for reengineering, where reengineering fits within the information management process, definitions of reengineering terms, and related standards and documents.

2.1 MOTIVATIONS FOR REENGINEERING

There are a number of reasons or motivations why an information system may need to be reengineered. These reasons range from a change in the functional process to the introduction of a new technology, with varying combinations of factors. Although there are a variety of reasons, below are at least three general situations that may initiate the consideration of reengineering:

- **New Functional Process:** A new functional process is created as a result of assessing and reengineering the functional process. New requirements may have been identified or new work flow patterns may be created, which may require changes to information system support. It is then the task of the information system department to determine how best to implement these changes.
- **System Upgrade Required:** In the second case, a problem with an existing system may have been identified (e.g., maintenance costs are rising too high). An analysis of the problem domain will then help to develop potential solutions to address this problem. Examples are the cost of hardware or software maintenance and the risk of inadequate hardware/software support. In this situation, the functional process has not changed or imposed new requirements upon the information system.
- **Technical Improvement Opportunities:** Technical improvement opportunities may also suggest potential changes in the functional process. These opportunities may be identified when conducting system reengineering, independent of functional (business) process reengineering.

These opportunities for system improvements only make sense if the functional process can be changed to exploit them.

2.2 SYSTEM REENGINEERING WITHIN THE INFORMATION MANAGEMENT PROCESS

The SRAM works in the context of the DoD's Information Management Process and the Center for Information Management (CIM) Software Systems Reengineering Process Model. This section discusses the relationship of the SRAM to those models.

2.2.1 Information Management Process Model

The DoD views information system (IS) support as an integral part of its Information Management Process (see Figure 2-1) [CIM94]. The decision to reengineer an information system will be done within a larger context of IS development, operation, and maintenance activities. This picture is a composite of functional and system activities, defining five major activities that include or interact with the reengineering process:

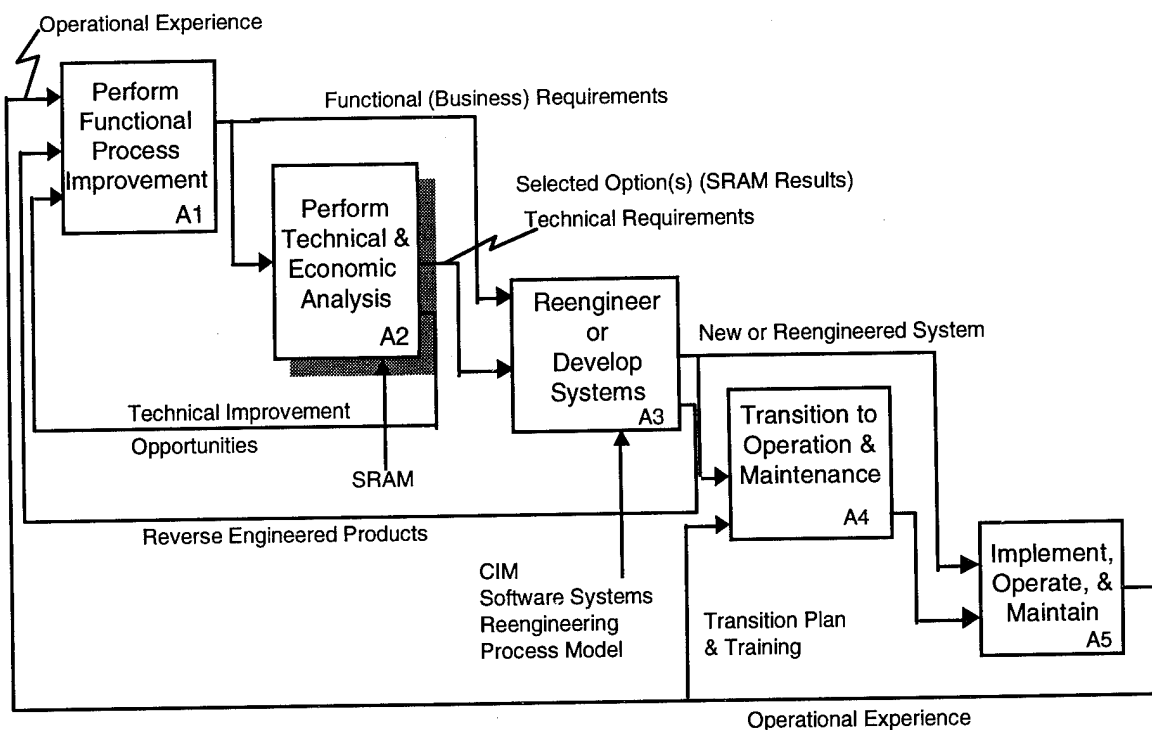


Figure 2-1. Information Management Process Model

- **Perform Functional Process Improvement:** This activity provides an analysis of the existing functional (business) process to determine areas for improving efficiency and reducing costs. These desired improvements are shown as new Functional (Business) Requirements. This activity also considers the effect of Operational Experience, Technical Improvement Opportunities, and Reverse Engineered Products. These requirements, including a functional process model, would affect any analysis and eventual selection of system engineering options.
- **Perform Technical and Economic Analysis:** The dual activities of technical and economic analyses of potential improvements to information system support are conducted to determine which system option will best support the new functional process. The SRAM would be applied primarily in this activity (shown as a mechanism in the figure). The SRAM results, Selected Option(s), would be an input to the next activity, Develop or Reengineer Systems. Technical Requirements and Technical Improvement Opportunities are also produced.
- **Develop or Reengineer Systems:** Within this activity, the new information system is built, either through new development or system reengineering. This activity uses the inputs of Selected Option(s) (the SRAM results), Technical Requirements, and Functional (Business) Requirements to produce the outputs of New or Reengineered System and Reverse Engineered Products. The CIM Software Systems Reengineering Process Model would also support this activity (shown as a mechanism in the figure).
- **Transition to Operation and Maintenance:** The deployment of the new information system is planned and preparations are made to incorporate it into the new functional process. This activity uses the inputs of New or Reengineered System and Operational Experience to produce the output, Transition Plan & Training.
- **Implement, Operate, and Maintain:** The new information system (which supports a new functional process) is put into operation where it undergoes normal use and maintenance upgrades. This activity uses the inputs, New or Reengineered System and Transition Plan & Training to produce the output, Operational Experience.

2.2.2 CIM Software Systems Reengineering Process Model

The CIM Software Systems Reengineering Process Model [CIM94] (see Figure 2-2) defines those activities necessary for reengineering an information system. Consisting of three major activities, Define Project, Reverse Engineer, and Forward Engineer, the model provides guidance and structure for conducting a system reengineering effort.

The information obtained from the SRAM would be used in the Define Project activity as a starting point for project definition. During the Define Project activity, the project would be planned in extensive detail, further defining objectives, identifying metrics and risks, and selecting appropriate tools and methodologies for development. The level of detail in the Reengineering Process Model, however, would go beyond that developed in the SRAM.

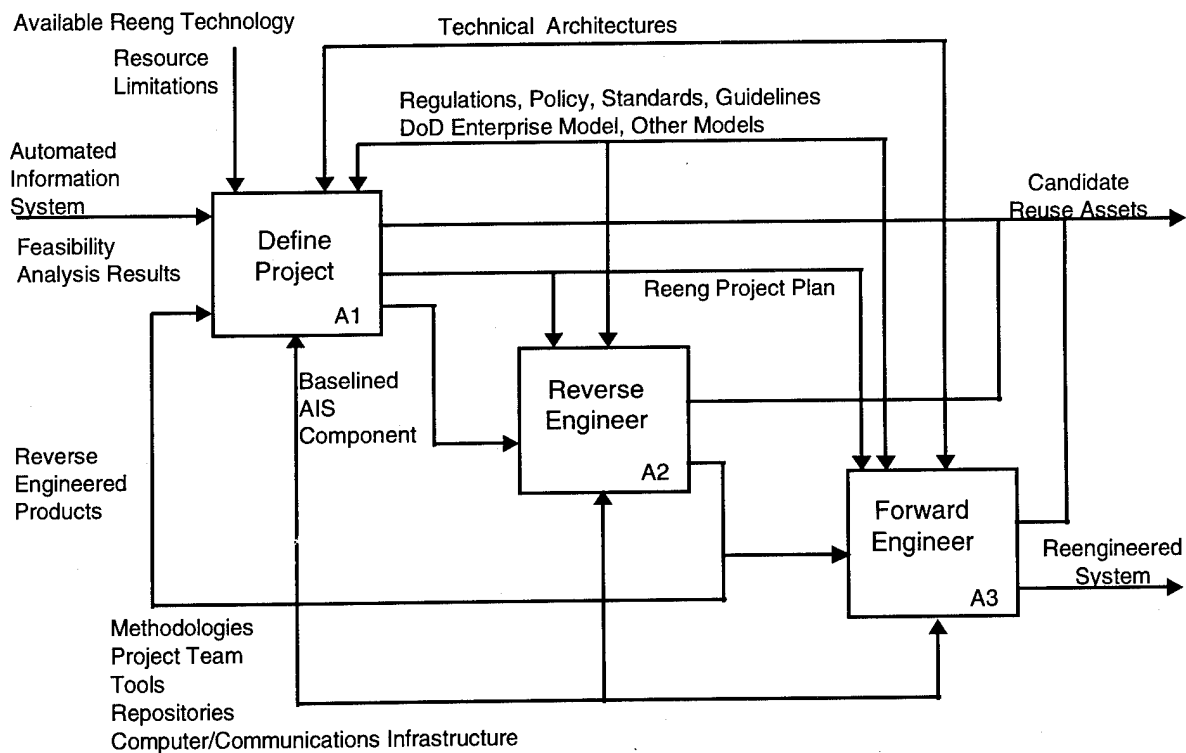


Figure 2-2. CIM Software Systems Reengineering Process Model

2.2.3 SRAM Context Diagram

Illustrated in Figure 2-3 is the Context Diagram of the SRAM. This picture illustrates the external inputs, controls, mechanisms, and outputs of the method. There are many other factors considered in assessing reengineering options; however, these factors, such as specific reengineering objectives, are developed internal to the method.

The SRAM takes an inclusive systems approach to assessing potential reengineering projects by considering all aspects of the project. This perspective includes the application software, data, technical infrastructure, and the interaction with the functional process. This approach tries to presume as little as possible about candidate projects and allows a combination of system strategies, such as a mix of new development and reengineering, to be applied. Note: It must be stressed that any reengineering effort should treat the SRAM as a set of guidelines; the SRAM is not a replacement for good sense, and it should be expected that there are areas within the SRAM analysis that will need greater or lesser degrees of attention. Section 3 provides a detailed description of the SRAM process.

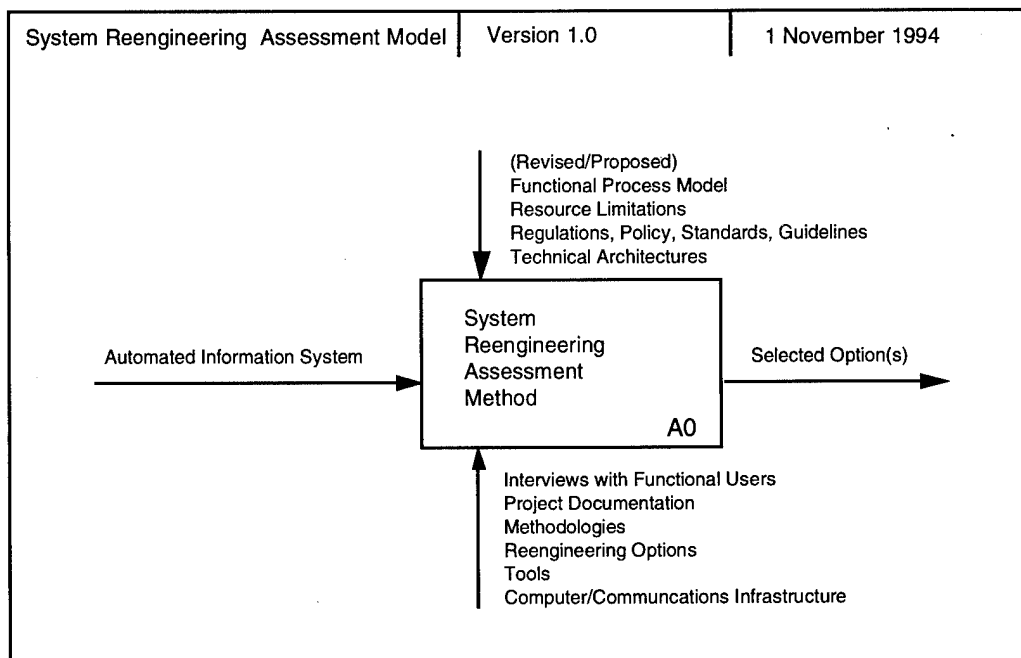


Figure 2-3. Context Diagram of SRAM

2.3 DEFINITIONS

Reengineering includes a variety of software engineering activities that range from simple code restructuring to full-scale reverse and forward engineering with new requirements. Below are definitions for the variety of reengineering activities. The Glossary includes these definitions as well as others related to software reengineering.

- **Software Reengineering:** Activities supporting the development and maintenance of automated information systems based on the examination and utilization of existing software system resources [CIM94]. The process encompasses a combination of other processes such as reverse engineering, restructuring, forward engineering, and translation. The goal is to improve the software system (functionality, performance, or implementation). Additional functionality is often incorporated into the system during this process [CIM93b, p. 3].
- **Reverse Engineering:** The process of examining an information system by analyzing its documentation, application software, and data structures within the environment in which the information system operates [CIM93b, p. 3]. This analysis is performed to (1) identify the system's components and their interrelationships, and (2) create representations of the system in another form or at a higher level of abstraction [CHI90]. The goal is to understand the existing software system (functions, performance, or implementation). Extracted information is represented in a format which can be integrated into the life cycle for development of a software system [CIM93b, p. 3].
- **Restructuring:** The transformation of a software system from one representation form to another, while preserving the external behavior both functionally and semantically [CHI90]. The goal is to improve the existing structure without altering the functionality [CIM93b, p. 4].
- **Forward Engineering:** Within the context of reengineering, forward engineering (consists of) the software engineering activities that consume the products of reengineering activities, primarily reverse engineering, reuse, and new requirements, to produce a target system [CIM94]. The goal is to create a software system via reengineering. This term primarily refers to the process of generating new software systems from reverse engineered designs. This term has evolved within reengineering to refer to those software engineering activities (traditionally performed during development) that are performed during or as a result of reengineering [CIM93b, p. 4].

- **Redocumentation:** Redocumentation produces supplementary information that provides understanding of the existing system and its components. This activity is usually performed to assist in (the) maintenance of existing systems. This activity does not alter the existing software system representation, nor does it generate any new representation to replace any part of the existing representation [CIM93b, p. 4]. Redocumentation is often performed as part of reverse engineering to produce interim documentation that is used to generate or is converted to reverse engineered products, (e.g., business rules, data models, and process models).
- **Translation:** Transformation of source code from one language to another or from one version of a language to another version of the same language. The goal is to improve the linguistic implementation of the software. This process is most successful when the two languages are similar or have a defined mapping between syntax [CIM93b, p. 4].
- **Software Reuse:** The application of existing software work products, including source code, documentation, designs, test data, tools, and specifications, in a software development effort other than the one for which each was originally developed. The goal is to facilitate the return on investment (ROI), improve software quality and reliability, shorten system development and maintenance times, increase productivity, and minimize software-related risks. Software reuse should be employed during reengineering and reengineering should be applied to identify candidate reusable assets [CIM93b, p. 4].

2.4 RELATED DOCUMENTS

Below is a list of documents that can provide additional information and guidance on software reengineering and cost analysis.

- *Functional Management Process for Implementing the Information Management Program of the Department of Defense, DoD 8020.1-M (Draft), August 1992 [DOD92].*
- *Center for Information Management Software Systems Reengineering Process Model, Version 2.0 (Draft), Defense Information Systems Agency, Joint Interoperability Engineering Organization, September 1994 [CIM94].*
- *Automated Information Systems Software Reengineering Risks Taxonomy Report, Defense Information Systems Agency, Joint Interoperability Engineering Organization, September 1993 [CIM93a].*
- *Information System Criteria for Applying Software Reengineering: Guidelines for Identifying Candidate Information Systems for Software*

Reengineering, Defense Information Systems Agency, Joint Interoperability Engineering Organization, May 1993 [CIM93b].

- *Center for Information Management Software Reengineering Project Planning Guide*, Version 1.0, Defense Information Systems Agency, Joint Interoperability Engineering Organizations, October 1993 [CIM93c].
- *Reengineering Economics Handbook, Proceedings of First Software Reengineering Workshop - Santa Barbara I*, Joint Logistics Commanders Joint Policy Coordinating Group on Computer Resources Management, March 1993 [JLC93].
- *User's Manual for the Functional Economic Analysis Model (Version 3.0)*, Institute for Defense Analyses, Alexandria, Virginia, December 1993 [IDA93].
- *A Descriptive Evaluation of Automated Software Cost - Estimation Models*, Institute for Defense Analyses, Alexandria, VA, October 1986 [IDA86].
- *Reengineering Technology Report*, Software Technology Support Center, Hill Air Force Base, Utah, August 1993 [STS93a].
- *Software Estimation Technology Report*, Software Technology Support Center, Hill Air Force Base, Utah, March 1993 [STS93b].

3. SRAM DETAILED DESCRIPTION

This chapter provides a detailed description of the SRAM. In the following sections, the context diagram (Figure 3-1) is divided into its component activities. The existing automated information system is the primary input to the SRAM and serves as the baseline against which any new options are measured. The SRAM is constrained by available resources, existing policy, standards, and architectures, and by the organization's functional process model. The functional process model, even in a proposed or tentative form, serves to set mission and user requirements that the information system must meet. Mechanisms to the SRAM are the functional users, existing system (with its documentation), and reengineering tools and methodologies. Finally, the SRAM produces a selected option(s) that best meets the full range of needs identified in the assessment.

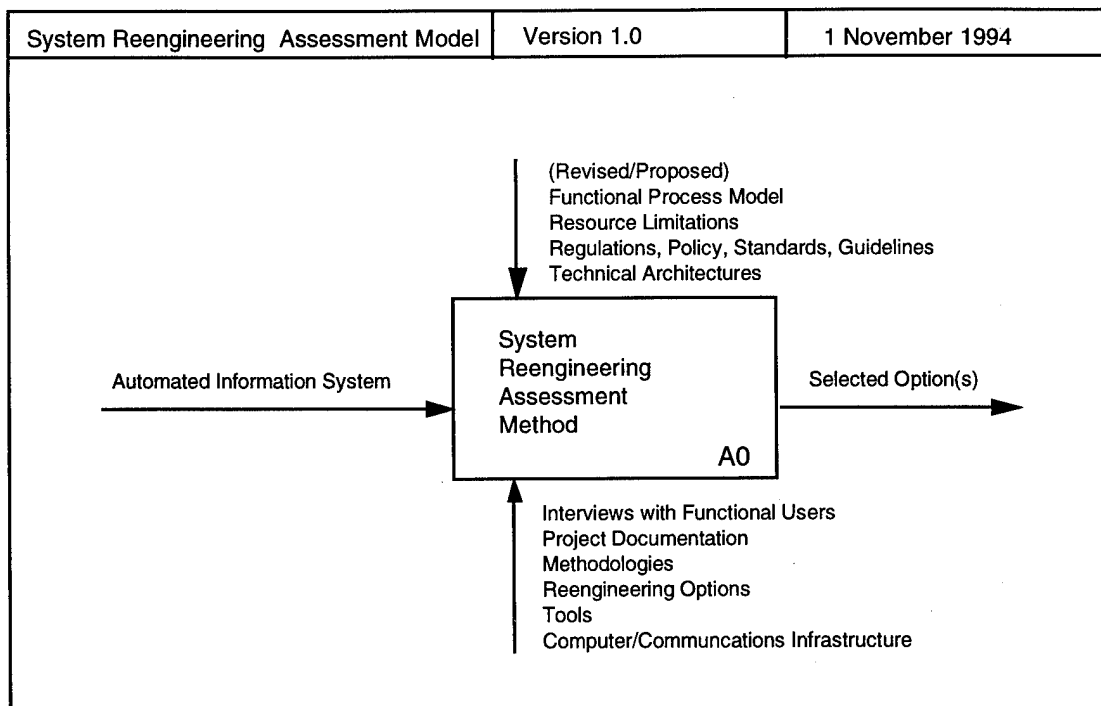


Figure 3-1. Context Diagram of SRAM

The SRAM (see Figure 3-2) consists of four major steps. Each major step, with its component activities, is described in the subsequent sections. For each activity, there is a description, list of ICOMs, a discussion of special considerations, and an example of the output. Where the SRAM activities may affect or interact with the CIM Software Systems Reengineering Process Model (or other CIM documents), this interaction is discussed in a footnote.

The SRAM consists of the following four steps:

- **Analyze Automated Information System and Environment:** All aspects of the problem domain that are to be considered by the reengineering effort are analyzed. During this step, objectives are determined, the effect upon the functional process considered, reengineering constraints identified, and a specific problem statement developed. This step is essential in ensuring that the reengineering effort solves the correct problem and considers its scope and limitations before developing possible system solutions.

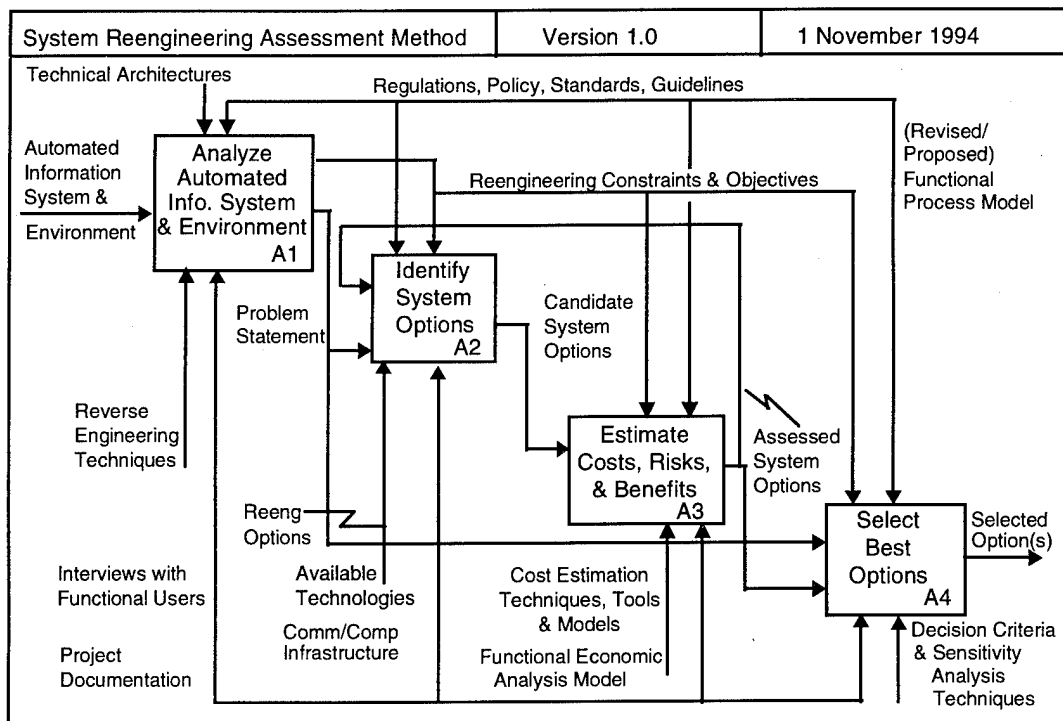


Figure 3-2. System Reengineering Assessment Method

- **Identify System Options:** System options are identified that satisfy the requirements and limitations of the problem. These candidate system options can consist of a mix of strategies that include new development, continued maintenance, reengineering, and system retirement. The method does not presume a single strategy approach for determining system choices. This step considers the problem requirements and the available software and hardware technologies to support the development of specific solution proposals that will be assessed for costs, risks, and benefits in the next step.
- **Estimate Costs, Risks, and Benefits:** Each of the candidate system options is assessed to its costs, risks, and benefits. The system cost assessment can also be done in the context of a larger Functional Economic Analysis (FEA). A variety of software cost models exist to assist the user in estimating costs. Risks are assessed with regard to planning, process, product, personnel, and technology. Benefits to be considered are those associated with system functionality, reliability, performance, usability, interoperability, compatibility, and maintainability.
- **Select Best Option(s):** The assessed system options are compared to one another, leading to a selection of the best system option(s). This selection is based upon the comparative assessments of costs, benefits, and risks for each system option. A business case for investment in software reengineering may have to be defended in terms of reducing future operation and maintenance costs, increasing the benefits provided to the mission, or reducing the risk of not delivering the services that the mission depends on. A number of decision criteria and decision making techniques are provided for comparing and selecting options.

3.1 ANALYZE AUTOMATED INFORMATION SYSTEM AND ENVIRONMENT (A1)

Objective The objective of the Analyze Automated Information System and Environment step is to investigate all aspects of the problem domain to be addressed by the reengineering effort. Analysis of the existing system is crucial because initial perceptions of functional and system problems are often incorrect or inadequate. This step tries to ensure that reengineering efforts do not fail because the wrong problem was addressed.

Activities This step consists of three activities (see Figure 3-3):

- *Conduct Problem Analysis* to isolate deficiencies or identify potential improvements in the current business computing system.
- *Identify Reengineering Objectives* to ensure that the potential reengineering effort is focused on solving the right problem.
- *Identify Reengineering Constraints* that will be imposed on candidate system solutions.

The Analyze Automated Information System and Environment step should not be confused with the CIM Functional Process Improvement activity. This activity does look at the existing functional process, but only to determine where information systems can be reengineered, thus enabling functional process improvements.

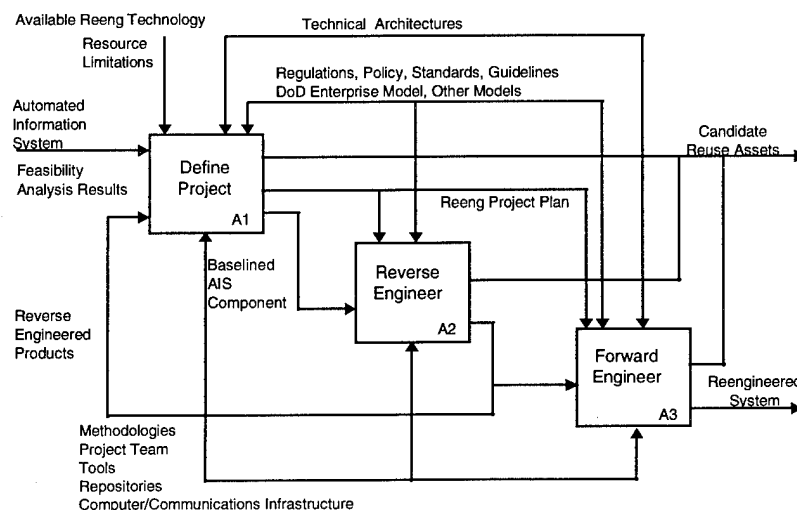


Figure 3-3. Analyze Automated Information System & Environment

3.1.1 Conduct Problem Analysis (A11)

Description This activity consists of examining the functional requirements of the computing system, observing how current system and functional processes satisfy these requirements, and identifying the costs associated with the existing system.

Inputs • Automated Information System and Environment

Controls • (Revised/Proposed) Functional Process Model
• Regulations, Policy, Standards, Guidelines
• Technical Architectures

Outputs • Problem Statement

Mechanisms • Interviews with Functional Users
• Project Documentation
• Reverse Engineering Techniques

Considerations Areas examined during this activity should include user requirements, current user satisfaction, technology employed, cost information and system criticality. The methods for obtaining this information may be simple or complex. Interviews with users and maintainers can be used to learn how the current system works, the technology employed, problems with its current operation, and desired improvements. More involved techniques, such as reverse engineering, may be needed to identify existing business rules that are undocumented. Financial records should be reviewed to identify system development, operation, and maintenance costs. Below are a number of areas that should be examined during the analysis of the existing system.

- **Functional Requirements.** One of the most important aspects of understanding a system is determining what the users really need the system to do. Some systems have the functional requirements captured in a requirements document. Often, however, this document is non-existent or out of date. These requirements are critical, because they may identify discrepancies between what is needed and what is currently implemented. Requirements must be examined to determine which are absolutely necessary, desirable, or nice to have. Interfaces with other systems should be scrutinized to ensure that the

information is needed and is being used, and that the proper type of information is being exchanged.

- **User Satisfaction.** Evaluating user satisfaction with a system is useful for identifying symptoms of problems. What aspects of the system do users find most useful? Least useful? An indicator of user satisfaction can be found by examining the system maintenance backlog. How long does it take to correct defects reported by users? What is the average length of time taken to correct defects reported by users? How long do some of the defects take to correct? If there isn't a maintenance backlog, are the users satisfied with the system, or is it not being used?
- **Technology Characterization.** If the existing system is over 5 to 10 years old, it may use antiquated technology. Potential problems may result from continued use and maintenance of this technology. Modern technology may provide better performance or functionality, or enable significant functional improvement.
- **Remaining System Life.** The expected remaining life of the system should be estimated. This "life expectancy" is essential in determining the most appropriate system strategy. A system with a short life expectancy (less than 5 years) will be judged differently than one with a long life expectancy (more than 15 years). The life expectancy will affect the length of time that returns on investment or benefits can be realized.
- **Costs.** Existing system costs should be explored in detail. Both operations and maintenance costs should be considered. It may be necessary to identify system costs for several years back, in order to establish a cost trend. Example costs include operations (system operators, support personnel, contract services, facilities, consumables, utilities); hardware/software maintenance; training; and program management.
- **Systems Criticality.** Does the system perform a service or set of services that are unavailable from any other source? Such a system would be vulnerable to risks such as schedule slippage or new technology introduction.

The following is an example of the output, Problem Statement, from the Conduct Problem Analysis activity. This example of the Aircraft Maintenance System is continued for each activity of the SRAM.

Example 3-1. Problem Statement

Problem Statement - A11

The U.S. Air Force Material Command is considering the reengineering of its Aircraft Maintenance System (AMS) for USAF cargo aircraft. This system is hosted on an Amdahl mainframe. The AMS is installed at 120 sites worldwide, and requires 5 full-time personnel to enter data, produce reports, and coordinate updates to other databases. The current system is 18 years old, consisting of 315 separate programs for a total of 620,000 lines of Cobol-74 code. The system was originally 250,000 lines, but has grown with changing requirements.

The AMS uses flat files and a command line interface; the data definitions are hardcoded within the programs. Data is entered manually from 17 different request forms, with no real-time verification of data entry and a resulting built-in delay of 72 hours for aircraft equipment retrieval.

Current hardware maintenance costs are \$100,000 (1994 dollars) per site for a total cost of \$12 million per year. However, this cost is expected to increase since the current hardware maintenance contract will expire this year, and costs will double in following years. Since this system is not part of a network, updates to the command-level database are provided on a monthly basis by shipping a 1600 bpi tape. The AMS also has its own data formats requiring a half-day conversion process when it uses data from other aircraft maintenance systems.

The existing AMS is now seen as an impediment to the aircraft maintenance functional process. The AMS currently requires five personnel at each site to support its daily operations: one data entry clerk, two system operators, one database administrator, and one system administrator. The delay in equipment retrieval has often resulted in equipment shortfalls, leaving a number of aircraft grounded. The proposed new functional process requires a maximum 12-hour delay in equipment retrieval, with on-line access to equipment catalogs, specification numbers, and descriptive information.

3.1.2 Identify Reengineering Objectives (A12)

Description This activity consists of identifying the types of changes that are needed to the existing system and to develop a set of reengineering objectives.¹

- | | |
|--------------------------|---|
| <i>Inputs</i> | • Problem Statement |
| <i>Controls</i> | • (Revised/Proposed) Functional Process Model |
| | • Regulations, Policy, Standards, Guidelines |
| | • Technical Architectures |
| <i>Outputs</i> | • Reengineering Objectives |
| <i>Mechanisms</i> | • Interviews with Functional Users |
| | • Project Documentation |

Considerations Once the existing information system has been thoroughly examined, it must be compared to the (possibly revised) functional process model to determine whether unfulfilled requirements exist. If there have been recent changes in the functional process, there may be many aspects of the existing system that need to be changed. If the motivation for reengineering stems from desired improvements in existing functional processes, then these desires should be quantified into a set of reengineering objectives.

Reengineering objectives should identify the problem and help scope the solution space, but should allow plenty of room for developing solutions. Ideally, the objectives should not directly focus on any one solution approach. For example, "Reduce by at least 50 percent the time required for equipment logging" identifies a problem without specifying a solution. The objective "Use bar code scanners" prescribes a solution without describing the problem. Rather than using bar code scanners, a more appropriate solution may be to eliminate the whole process of equipment logging. Thus, while the use of bar code scanners may improve the existing functional process, a much better solution

¹While similar to the "Identify Objectives" activity in the "Define Project" activity of the CIM Software System Reengineering Process Model [CIM94], the objectives described in this section are more general and systems oriented as opposed to project specific.

may develop if the solution space is not constrained too early. As another example, a revised functional process may require new procedures for performing inventory. The system that supports inventory will need to be modified to reflect the new procedures. The new requirements should be included within a set of reengineering objectives, but the specific modifications should not be detailed.

Representative objectives that might be developed during this step include the following:

- Reduction in the time necessary to complete a task (e.g., the aforementioned "Reduced by 50 percent the time required for equipment logging," and "Provide reports to users within 24 hours of their request").
- Elimination of processes (e.g., "Eliminate all re-keying of data once it has become available on-line").
- Reduction in the costs associated with a system (e.g., "Modify our systems so that we reduce the number of system operators required by 2.5 percent, yet sustain existing level of services," and Reduce the annual expenditures on system maintenance by 2.5 percent while sustaining existing level of services").
- Improvement in performance and quality (e.g., "Provide interactive response to users queries" or "Reduce the number of out-of-date data fields reported by users by 90 percent").

An interesting example of initially focusing on improper reengineering objectives was recently reported by Jeff Moad [MOA93]. A chemical company desired to reduce the time needed to reimburse employees for expense accounts. The existing process required a number of different forms for signature, as well as authorization from several departments. The information system department automated the existing process by installing a local area network, allowing department managers to more efficiently pass the various forms around electronically. Although the project was a technical success, it failed because it didn't consider

changing the business (functional) process. (Often when a problem is encountered, the best solution involves developing a different process, not simply automating the existing process). Continuing this example, the company ended up developing another solution from scratch, providing better results. Only after changing the business process did the project address the real objective: significantly reduce the number of forms used by the various departments.

Example 3-2. Reengineering Objectives

Reengineering Objectives - A12

Objectives of reengineering the AMS are 1) to reduce the delay in equipment retrieval from inventory by 50% (may require real-time verification of equipment requests), 2) to meet the real-time needs of the new command-level acquisition process, and 3) to reduce annual operational costs of the AMS by at least 10 percent.

3.1.3 Identify Reengineering Constraints (A13)

Description This activity consists of identifying any constraints on potential solutions that may exist due to organizational policy (e.g., all new software must be developed in Ada), budgetary considerations (e.g., the reengineering effort is limited to X dollars), and technical requirements (e.g., the old system must be operated for six weeks after installation of the new system). The set of constraints must be identified early, in order to assist in developing viable reengineering options.²

- | | |
|-------------------|---|
| Inputs | • Reengineering Objectives |
| Controls | • Resource Limitations |
| | • Regulations, Policy, Standards, Guidelines |
| | • Technical Architectures |
| | • (Revised/Proposed) Functional Process Model |
| Outputs | • Reengineering Constraints and Objectives |
| Mechanisms | • Interviews with Functional Users |
| | • Project Documentation |

Considerations Often, two or more constraints will contradict each other. Trade-offs must be made to resolve these contradictions. In addition, every constraint should be examined to determine whether it is truly a constraint or actually a desired objective. For example, a reengineering effort may be limited to X dollars. However, if twice the functionality could be delivered for 10 percent additional investment, the organization should reconsider the constraint.

Example 3-3. Reengineering Constraints

Reengineering Constraints - A13

The new AMS should conform to the standards designated in the DoD Technical Reference Model (TRM) and the USAF data element definition standards. These constraints mean that any new software development (if more than 30% new code) will be in Ada 83; the data base access method will be SQL, and the host operating system will be POSIX compliant.

²The results of this activity can be used in the Reengineering Project Plan as defined in the *CIM Software Systems Reengineering Project Planning Guide* [CIM93c].

3.2 IDENTIFY SYSTEM OPTIONS (A2)

Objective The objective of this step is to generate specific system options that address the problem, objectives, and reengineering constraints from the previous step. A single solution may seem “obvious” from the problem statement. There is usually more than one way to solve any problem and the first one found is not always the best. At least two or three technically feasible options should be outlined as a result of this step.

Activities This step consists of three activities (see Figure 3-4):

- *Identify System Reengineering Strategies* to establish a basis for constructing specific system solutions.
- *Assess Technology Alternatives* to determine those information technologies appropriate to be applied to the potential reengineering.
- *Develop System Configurations* to provide specific options that can be assessed according to comparative costs, risks, and benefits.

The estimation of costs, risks, and benefits is the subject of the next activity, A3. It is not intended, however, that identification of options in this step be completely independent of these factors. In fact, it is essential to consider the trade-offs between cost, benefit, and risk when seeking options and other solutions.

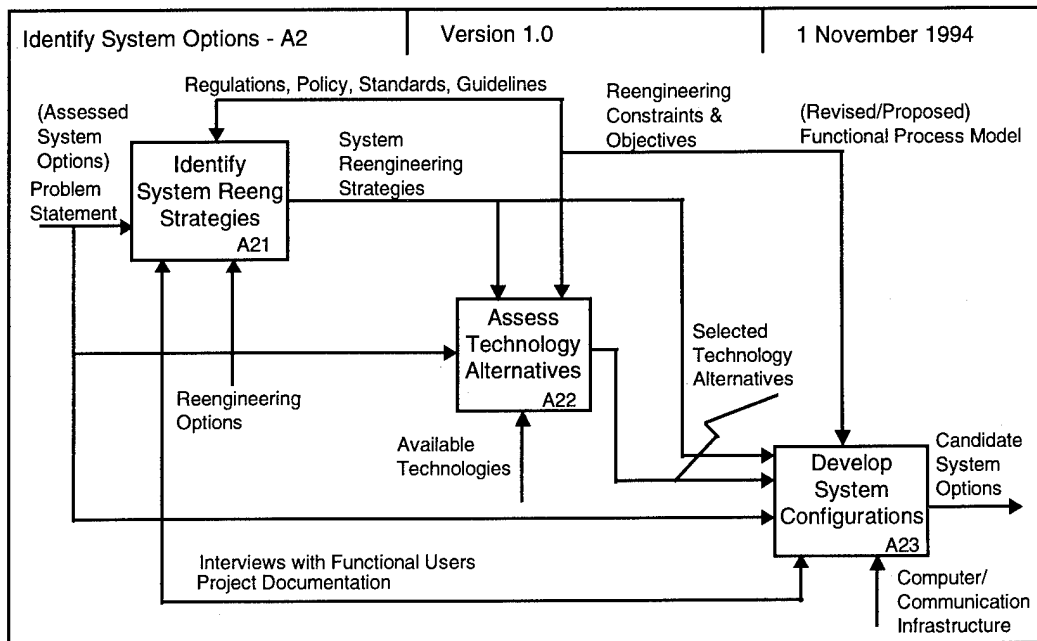


Figure 3-4. Identify System Options

3.2.1 Identify System Reengineering Strategies (A21)

Description This activity consists of identifying a number of possible reengineering strategies. Selection of a reengineering strategy is needed as a basis for technology selection and development of candidate system options.³

- | | |
|--------------------------|---|
| <i>Inputs</i> | <ul style="list-style-type: none">• Problem Statement• (Assessed System Options) |
| <i>Controls</i> | <ul style="list-style-type: none">• Reengineering Constraints & Objectives• Regulations, Policy, Standards, Guidelines• (Revised/Proposed) Functional Process Model |
| <i>Outputs</i> | <ul style="list-style-type: none">• System Reengineering Strategies |
| <i>Mechanisms</i> | <ul style="list-style-type: none">• Reengineering Options• Interviews with Functional Users• Project Documentation |

Considerations There is a relatively broad spectrum of possible strategies that are considered within the realm of system reengineering. While encompassing the application, data, and infrastructure domains, these strategies can have elements from a mix of reengineering options. These reengineering options include forward engineering, reverse engineering, new development, continued maintenance, code translation, redocumentation, data reengineering, and commercial-off-the-shelf (COTS) software and hardware acquisition.

The scale of the reengineering effort will depend upon the problem to be solved, the state of the existing system, the technology to be employed in the new system, and available resources (including schedule).

System options that have been previously assessed in the SRAM may also be used as input to this activity. Options may need to be revised or refined based upon analysis or information obtained

³This activity is similar to the CIM Software Systems Reengineering Process Model [CIM94] activity entitled "Developed Reengineering Strategy" (A131), which commences with the definition of the reengineering project plan. Since the SRAM precedes the Reengineering Project Plan Definition, outputs of this activity could be used as inputs to the Project Plan.

in subsequent SRAM activities.

Example 3-4. System Reengineering Strategies

System Reengineering Strategies - A21

Baseline: Continue with current system through existing maintenance. Basically leave system as is; budget for increased hardware maintenance costs; and reconsider when (and if) a new functional process is finally approved.

Option A: Upgrade current system through existing maintenance. Add capability to do real-time verification of equipment requests and budget for increased hardware maintenance costs. Some reverse engineering will be necessary to add verification capability. This option allows for the elimination of the data entry. So a total of 4 support staff are required at each site for this option. Although this option is not RAFIM compliant, it will be considered as an interim solution. It is recognized that this option has a maximum lifetime of 6 years until which time TAFIM compliance cannot be waived.

Option B: Reengineer existing system to include real-time verification of equipment requests, move to new platform, database, and software implementation that complies with USAF and DoD architecture requirements. This option involves reverse engineering, forward engineering, some new development, and redocumentation. This option eliminates the need for the data entry and one system operator, requiring a total of 3 support staff to support the system at each site.

3.2.2 Assess Technology Alternatives (A22)

Description This activity consists of assessing current and emerging technologies for possible use in the reengineering effort. It should be noted that assessing technologies may result in ideas for additional capabilities for the eventual system and even changes in the functional process.⁴

- | | |
|--------------------------|---|
| <i>Inputs</i> | <ul style="list-style-type: none">• Problem Statement• (Assessed System Options) |
| <i>Controls</i> | <ul style="list-style-type: none">• Reengineering Constraints & Objectives• System Reengineering Strategies• (Revised/Proposed) Functional Process Model• Regulations, Policy, Standards, Guidelines |
| <i>Outputs</i> | <ul style="list-style-type: none">• Selected Technology Alternatives |
| <i>Mechanisms</i> | <ul style="list-style-type: none">• Available Technologies |

Considerations The advancement of technology since the development of any 10-year or older legacy system provides a vast array of potential solutions for reengineering. Although leading-edge technology should be viewed as somewhat of a risk, any reasonably mature modern technology that addresses the problem should be considered a potential candidate. This should include any technology that would be considered for use in starting a new system. Reengineering a 15-year-old system with 10-year-old technology should probably be viewed as counterproductive.

The following discussion is intended to stimulate consideration of possible approaches to information system problems. Since specific technologies can quickly become dated, consideration should not be limited to those mentioned here.

Hardware Technology

- Processor speed and memory continue to double every two to three years and prices continue to drop. Networking and distributed computing are growing almost as fast. This has such

⁴ This activity can also provide input to the "Identify Tools and Methodologies" activity within the "Reengineering Project Plan Definition" node of the CIM Software Systems Reengineering Process Model [CIM94].

a significant effect on software that it is increasingly difficult to make decisions about hardware and software separately. While it is possible to reengineer only the software component of a system, planning for hardware and software upgrades should be fully coordinated.

- New computing hardware offers a high degree of performance and connectivity. Personal computers and workstations provide sophisticated user interfaces with high-resolution graphics and immediate interactive responsiveness for most local processing. High-speed, local area networks can connect these devices to file servers, conventional mainframes, and “outside” data and computing resources via wide area networks. This flexibility can be employed to extend existing systems and to off-load processing from older, overworked mainframes. Changes in the way computing resources are used include on-line access to data, reducing the number of printed reports, electronic transfer of data, and reducing the handling/shipping of exchange media such as magnetic tapes.

Software Technology

Commercial off-the-shelf (COTS) software products:

- **Relational Database Management System (RDBMS):** Modern relational database management systems provide extensive facilities for organizing data, producing routine reports, and answering ad hoc queries.
- **Graphical User Interface (GUI) Generator:** Tools for developing and maintaining graphical user interfaces can make complex software systems easier for people to operate productively.
- **Hypermedia:** Tools for developing and maintaining hypermedia can handle information in a variety of forms and with arbitrary associations to other stored information.
- **Multimedia:** Information can be represented in a variety of forms beyond typical data formats. Multimedia brings together graphics, audio, video, and animation for coordinated, interactive use.

In-use or in-development DoD software:

- **Reuse Repositories:** Stockpiles of software requirements, designs, components, and related information can be searched by keywords and other criteria.

New development: Some new software may be necessary, however, to convert from the existing system to COTS or reusable systems, or to "glue" together complete systems from COTS or reusable components.

- **Ada:** Beyond being required by the DoD, Ada is the programming language of choice for data conversions and combining reusable software components to form new systems. DoD software repositories encourage the development and reuse of Ada components. Ada provides extensive support for data type definition and encapsulation, strong type checking, exception handling, and component packaging.
- **4GLs:** Fourth generation programming languages are supported by many COTS database management systems. These languages are tailored for data processing applications and often yield much higher productivity than conventional programming languages.
- **SQL:** SQL is a standard query language that was developed around the relational database model. Most relational database management systems and some network database systems support SQL as the basis for generating routine reports and for ad hoc queries.
- **Object-Oriented Technology (OOT):** OOT offers several advantages for developing reusable software components over conventional software development techniques. Many programmers find the object-oriented perspective more intuitive than the conventional methods of functional decomposition. The type class mechanism enforces common definitions of operations for member types. New data types derived by extending existing types can often inherit operations from the original definition and potentially avoids various levels of recoding.
- **Client-Server:** Distributed processing based on the client-server model and standard protocols can provide flexible, scalable solutions to many information management problems.

Example 3-5. Selected Technologies

Selected Technologies - A22

A variety of technologies were considered, including client/server, relational DBMS with SQL, multimedia, Ada, 4GL, Object-Oriented Technology, and GUI.

The selected technologies for the reengineering option (Option B) were as follows:

- 1) Client/server (will allow for distributed computing).
- 2) Ada (compliance with DoD policy, opportunity for reuse within the functional domain, and use of object-oriented design).
- 3) Relational DBMS with SQL (TRM) compliant and more mature than Object-Oriented DBMS).
- 4) X/Motif GUI (TRM compliant).
5. Hardware will consist of COTS components; however, it should support the above software technologies and access to the on-site LAN.

Technologies not selected were 4GL (not sufficient capability), multimedia/hypermedia (although not ruled out for the future), and object-oriented DBMS (not sufficiently mature and no current access standards.)

3.2.3 Develop System Configuration (A23)

Description This activity consists of developing specific system configurations in sufficient detail so that each can be assessed later with regard to costs, risks, and benefits. Each system configuration with its reengineering strategy will form a candidate system option.⁵

- | | |
|--------------------------|---|
| <i>Inputs</i> | <ul style="list-style-type: none">• Problem Statement• (Assessed System Options)• Selected Technology Alternatives• System Reengineering Strategies |
| <i>Controls</i> | <ul style="list-style-type: none">• Reengineering Constraints and Objectives• (Revised/Proposed) Functional Process Model• Regulations, Policy, Standards, Guidelines |
| <i>Outputs</i> | <ul style="list-style-type: none">• Candidate System Options |
| <i>Mechanisms</i> | <ul style="list-style-type: none">• Computer/Communications Infrastructure• Interviews with Functional Users• Project Documentation |

Considerations Each of the proposed system options to solving the stated information system objectives should be described in sufficient detail to enable the cost estimation, evaluation, and decision making steps that follow in this process. Development, testing, and transition plans, including schedules, should be outlined in each option.

System options should identify all the hardware and software components involved in or affected by the solution approach, including old components that are being removed or modified, as well as new components that are being introduced. System configurations before and after the proposed changes should be shown.

COTS software components and sources of reused software components should be identified. The support available for reused

⁵ When the candidate system option is chosen and the system enters the reengineering phase described by the CIM Software Systems Reengineering Process Model [CIM94], the system configuration developed under this activity can provide input to the model's Project Definition activity.

components should be noted. Size estimates of all reused and custom-developed software should be given. The sizes of database schemas, report specifications, user interface screen definitions, and other tool inputs should be included in these estimates. In addition, the extent of any changes to COTS or reused software that might be necessary should be estimated.

Example 3-6. Candidate System Options

Candidate System Options - A23

Baseline: Keep current AMS with existing platform, operating system, and functional process. Software maintenance costs will remain as projected; hardware maintenance costs will double, with a 10 percent increase per year in following 4 years. Operations costs will remain as projected.

Option A: Keep existing platform; add real-time data verification. This option eliminates one position per site. Software and hardware maintenance costs are the same as Baseline. To add the data verification, 6 programs (approximately 23,000 lines of code (LOC)) will need to be reverse engineered and restructured.

Option B: Reengineer from Cobol-74 to Ada83. This option would require reverse engineering 143 programs, approximately half of the code (300,000 LOC), to extract current operational functions and data elements (5060 defined within 65 programs - 200,000 LOC). The remaining programs are either no longer used or easily replaced with COTS. Software would consist of Sun Solaris operating system, Oracle relational DBMS and SQL, TCP/IP, a mail tool, X-Windows, and document preparation software. Each site would have five Sun color X terminals with 24 MB of RAM, one SPARC-10 server with 8-GB disk storage, 5-GB tape drive for backups, and two 150-MB tapes drives.

3.3 ESTIMATE COST, RISKS, AND BENEFITS (A3)

Objective The objective of this step is to determine the costs, risks, and benefits for each candidate system option. This step separates the estimation of system costs, risks, benefits and functional costs into distinct activities and does not require that all four assessments be performed.

Activities This step consists of four activities (see Figure 3-5):

- *Identify and Estimate Risks* to determine the risks associated with planning, development, personnel, product, and technology.
- *Identify and Estimate Benefits* to determine the benefits with regard to system functionality, reliability, performance, and usability.
- *Identify and Estimate System Costs* to determine the system costs for application, data, and infrastructure reengineering.
- *Identify and Estimate Functional Costs* to determine the overall economic effect to the functional process.

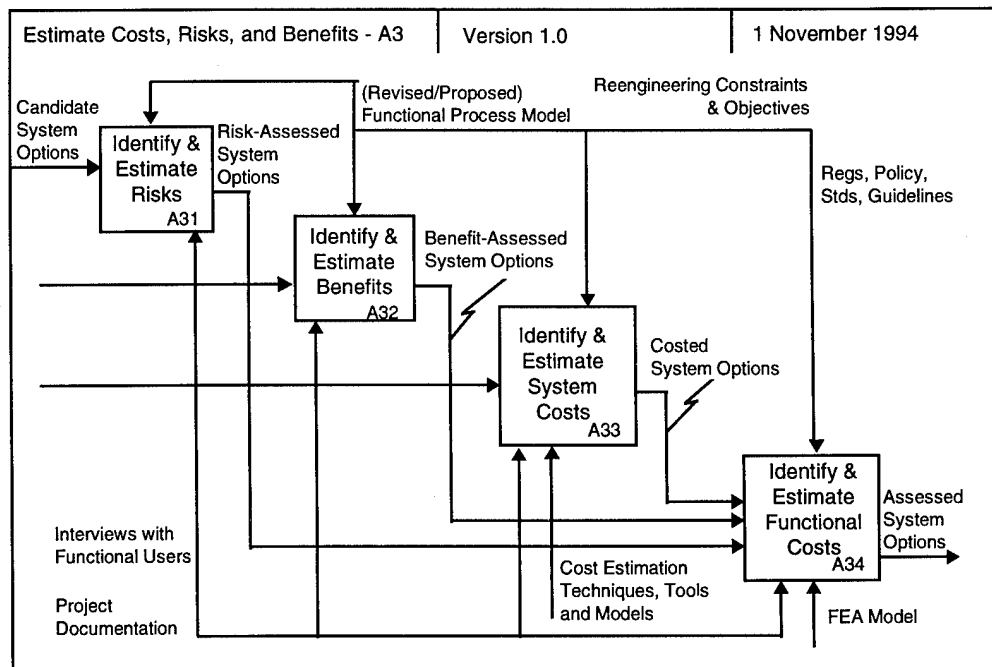


Figure 3-5. Estimate Costs, Risks and Benefits

3.3.1 Identify and Estimate Risks (A31)

Description This activity consists of identifying and estimating the risks associated with each system option.

Inputs

Controls

Outputs

Mechanisms

- Candidate System Options
- Reengineering Constraints & Objectives
- (Revised/Proposed) Functional Process Model
- Regulations, Policy, Standards, Guidelines
- Risk-Assessed System Options
- Interviews with Functional Users
- Project Documentation

Considerations While there is no universally accepted definition of risk, for our purposes, risk can be considered the potential for incurring harm or loss. Within software engineering, the identification and estimation of risk is still a developing discipline. DISA has developed a risk taxonomy (see below) for the identification of risks within a software reengineering effort. However, other sources may be useful in understanding and estimating software-related risks. See [BOE89, CAR93, CHA88, KIR92] for additional software risk background.

In some cases, it may be possible to express a risk in terms of cost or cost savings, and these quantifications can be factored into the cost assessment for each system option. However, to convert a risk into cost may not be the most desirable option for expressing that risk. There are decision making techniques that allow comparisons between different types of factors; these techniques are discussed in Section 3.4, with an example in Appendix A. See [CHI89, MOR90, RAI68] for further information on risk rank techniques.

It is often easy to think only of those risks associated with the system reengineering options. However, in assessing risk, we should also consider the risk of non-action, i.e., remaining with the status quo system and functional process. For assessing system options, the status quo will usually be the baseline option.

Software Reengineering Risk Categories

For the identification of risks, there are five categories of risk that have been defined in the Automated Information Systems Software Reengineering Risk Taxonomy Report [CIM93a]: 1) Planning, 2) Process, 3) Personnel, 4) Product, and 5) Technology.

- **Planning:** Planning risks are those associated with effort, schedule, and costs, and includes the selection of the reengineering strategy, approach, and reengineering goals such as maintainability, reliability, portability, and the ability to incorporate new functionality.
- **Process:** The process risk category includes those risks with the software engineering activities associated with a reengineering effort, such as goal establishment, project organization, plan development, technology selection, and team building.
- **Personnel:** The personnel risk category includes those risks with personnel availability, general knowledge / training / experience, application domain expertise, program / system knowledge, reengineering expertise, motivation, and team composition.
- **Product:** The product risk category includes those risks associated with the characteristics of the existing and target system, such as the complexity and quality of the software requirements, design, source code, documentation, host/target platforms, available reengineering and development tools, required standards, data models and quality, existing system age, intended system longevity, and the effect and importance to the enterprise.
- **Technology:** The technology risk category includes those risks associated with the methods and tools of reengineering, specifically their capabilities, availability, maturity, appropriateness for the application, level of automated support, and reusability of artifacts.

Example 3-7. Risk-Assessed System Options

Risk-Assessed System Options - A31

Baseline: There are virtually no risks related to software reengineering since this option doesn't include any reengineering of the system or functional process. There is little to consider in terms of planning, process, personnel, and technology. The problem statement outlines a number of problems (risks which have materialized) and risks of staying with the current system and functional process. A major risk is the sustainability of the current hardware system (which will also double in maintenance costs) and hence the sustainability of the functional process. Product system reliability and service availability both constitute areas of substantial risk as noted in the problem analysis.

Option A: As with the Baseline Option, the major risk is the sustainability of the current hardware platform. Service availability will have improved under this option with its associated improvements in the functional process. Reengineering risks are relatively minor since the reengineering effort will use the same language for application upgrades. No new development personnel will be required. Thus, there is still relatively little risk for the planning, process, and technology of Option A.

Option B: This option has the "highest" software reengineering risk since it contains the most reengineering effort. Because of the extensive reengineering activities, planning and process risks are a major consideration and will need to be addressed if this option is chosen. Although the technologies chosen are fairly mature, new expertise and training will be required to accommodate the language change from Cobol to Ada the use of an RDBMS. Software development costs should reflect the need for additional training and the hiring of specialized Ada and RDBMS expertise. On the positive side, the new system will be far more reliable than either the Baseline or Option A. Hardware sustainability is virtually assured since it would come under normal vendor maintenance agreements. The finished system will also support the new functional process, thus process, thus reducing the risks of non-sustainability of the functional process.

3.3.2 Identify and Estimate Benefits (A33)

Description This activity consists of identifying and estimating any benefits associated with each system option.

- | | |
|-------------------|---|
| Inputs | • Candidate System Options |
| Controls | • Reengineering Constraints & Objectives |
| | • (Revised/Proposed) Functional Process Model |
| | • Regulations, Policy, Standards, Guidelines |
| Outputs | • Benefit-Assessed System Options |
| Mechanisms | • Interviews with Functional Users |
| | • Project Documentation |

Considerations In identifying and estimating benefits, it will be necessary to consider the benefits to both the business and system. As noted previously with risk, it may be possible to express benefits in terms of cost savings. However, making that determination, i.e., converting a benefit into cost savings, may be quite difficult and produce results that are questionable. Some of the types of benefits to consider are the following:

- **Functionality:** Breadth (scope) and depth (level of detail) of system functionality and the capabilities provided to the overall functional process.
- **Reliability:** Capability to perform as expected (without error) and the capability to handle anomalous conditions. This benefit should be assessed in both the system and functional process.
- **Performance:** System and functional process performance in terms of response time, processing capability, numbers of transactions, and processing time.
- **Usability:** Ease of use for system and functional users.
- **Interoperability:** Interoperability with other systems and data.
- **Compatibility:** Compatibility with existing standards and architectures.
- **Maintainability:** Quality of design and implementation, e.g.,

not overly complex and highly portable.

Example 3-8. Benefit-Assessed System Options

Benefit-Assessed System Options - A32

Baseline: The most notable benefit of the Baseline is that this system is already in place with supporting systems and personnel. Otherwise, there are no benefits with regard to reliability, data interoperability, support for new functional process, or compatibility with USAF standards. This system is also difficult to use with its existing command line interface.

Option A: Option A provides for a higher degree of reliability and shorter request processing time with the real-time verification of entry data. This improvement in efficiency should eliminate the need for one full-time system support person. This option also has few benefits in terms of usability and compatibility.

Option B: Option B provides the highest benefits in terms of functionality and reliability, data and application interoperability. This option also provides the means to upgrade the current functional process, saving a maximum of two full-time support personnel. Other benefits include increased portability (from being TAFIM compliant). With an X/Motif GUI this system will be more user friendly than the Baseline or Option A.

3.3.3 Identify and Estimate System Costs (A33)

Description This activity consists of estimating the costs for system development, maintenance, and operation for each candidate system option. In assessing the system costs, the system is divided into the general areas of application software, data, and infrastructure.

- | | |
|-------------------|---|
| Inputs | • Candidate System Options |
| Controls | • Reengineering Constraints & Objectives |
| | • (Revised/Proposed) Functional Process Model |
| | • Regulations, Policy, Standards, Guidelines |
| Outputs | • Costed System Options |
| Mechanisms | • Cost Estimation Techniques, Tools, and Models |
| | • Interviews with Functional Users |
| | • Reviews of Program Documentation |

Considerations The reengineering of a system will entail a number of costs beyond the immediate software application itself. Any data used by the legacy system may need to be reengineered, along with changes to the hardware platform and operating system.

General Cost Factors

There are a number of cost factors that apply to nearly all situations.¹

- **Size of Effort:** Size has a major effect upon the application and data reengineering, whether the number of staff months, lines of code, function points, data elements, or documentation pages.
- **Number of Sites:** The number of installed sites for the reengineered system will affect any component that is acquired either through COTS or NDI (non-developmental item), particularly hardware and common application software such as spreadsheets or DBMSs.
- **Experience and Expertise of Staff:** If current maintenance personnel are to conduct the reengineering, training may be necessary, particularly to incorporate Ada, relational databases, and client/server capabilities. Otherwise, estimate costs for bringing in

¹ For further discussion of potential cost factors, see *Information System Criteria for Applying Software Reengineering* [CIM93b] for its review of product characteristics and process factors of existing and reengineered software systems.

new expertise, either government or contractor.

- **Available Tools:** New tools and support environments will in most circumstances need to be acquired. Anticipate acquiring tools to support code development (compilers, editors, debuggers, etc.), reverse engineering, data base design, application software analysis and design, and documentation.
- **Access to Functional Experts:** Functional experts may not be local and their knowledge will be essential. Costs estimates should include any travel that may be necessary.
- **Quality:** Quality is a factor for both the old and new systems: e.g., the old system may have very complex code and be difficult to reverse engineer; the new system may require high reliability and security.
- **Transition Costs:** It may be necessary to keep two systems operational until the new or upgraded system is in place. Other transition costs include training for functional users and system support staff.
- **Life Expectancy:** Determining comparative returns on investment should consider the life expectancy of the specific functional process, the system, and components of applications, data, and infrastructure.

Specific Cost Factors

The following section divides system costs into the areas of application (mission- or functional-specific software programs and specifications); data (legacy data, data specifications, and data management software); and infrastructure (hardware, communications, and system-level software).

Application The major concern here is with the various activities within software reengineering and each may be costed differently. It is expected that the reengineering effort will have a combination of these activities and each activity should be appropriately costed. COTS software should be calculated on a per site basis, including any annual maintenance costs. Table 3-1 summarizes the considerations and cost techniques for the various activities in Application Reengineering Cost Estimation.

Note: Not all software costs models cover the same portions of the life cycle. To determine which cost model would be most appropriate, the user should consult current sources of software cost models and tools. See [IDA86], [BOE81], [WEL92], and [GUL93] for further background regarding software costing.

Table 3-1. Application Reengineering Cost Estimation

| Activity | Considerations | Cost Techniques |
|-----------------------|--|--|
| New Development | A new development presumes new functional requirements, a full requirements analysis phase and a set of new documentation. The costs per LOC may also vary according to whether the code is to be reusable. Data from the USAF Standard Systems Center (SSC) shows a \$66 development cost per line of reusable Ada code compared to a range of \$4 to \$20 per line if reusability is not considered [SSC93]. | Software cost estimation models or specific estimates of labor costs. |
| Continued Maintenance | Changes will occur to the existing system in its current form. These changes reflect the typical corrective and adaptive maintenance that occurs within the "normal" maintenance budget. Special efforts to reverse engineer, restructure, or redesign the software, particularly for enhanced future maintainability may really be reengineering activities and should be costed accordingly. | Current cost projections and staffing estimates can be extra-polated to determine the cost for continuing maintenance on the existing application. |
| Reverse Engineering | Estimating the cost will depend heavily upon the products desired from reverse engineering. Such products include design and requirements specifications. Tools are available for extracting design structures from legacy code. Identifying mission and system requirements will be more difficult and will probably require the involvement of functional users. Understandability of code and documentation will also affect the required effort. | No common cost models currently exist. Use specific estimates of labor costs. |
| Forward Engineering | This activity presumes that requirements are derived from an existing system, either from existing specifications or reverse engineering. Software cost models or work breakdown structures are appropriate to use as long as the model is properly scoped to the effort. For example, all forward engineering efforts do not begin with requirements analysis. Some may eliminate the requirements analysis step, using the existing specification and proceed with a new design. Some cost models can accommodate these shortened life cycles. If using a software cost model, a change in source code language also needs to be considered in the estimates for size and for potential reuse. SSC estimates \$12 per LOC (Ada) for "technical modernization" where no new functional requirements are included [SSC93]. | Use software cost estimation models or specific estimates of labor costs. |
| Restructuring | This activity restructures and optimizes existing code implementations. There may be an effect upon the detailed design which may require changes to existing documentation. | Estimates of labor costs. |
| Code Translation | The differences between existing and target languages will be a major cost factor. For simple code translation such as between different versions of Cobol, SSC estimates \$0.28 to \$1.82 per line of code [SSC93]. | Cost models, with lines of code estimates. |
| Redocumentation | This activity will depend upon the standards applied, the life-cycle phases, and any tailoring of the documents. Also consider tools to support automatic document generation. | Estimates of labor costs. |

Data

Information system reengineering will likely have a large data component; in some cases, it may be that only the data is retained from a legacy system. This area includes the DBMS (which may be integrated within the existing application), data models, data element descriptions, and data dictionaries. Table 3-2 summarizes data reengineering cost estimation considerations and techniques by activity. Note that the cost for any COTS, such as a DBMS, will need to be multiplied by the number of installed sites. COTS cost estimations should also include costs for maintenance.

Table 3-2. Data Reengineering Cost Estimation

| Activity | Considerations | Cost Techniques |
|---------------------|--|---|
| New Development | New functional and data requirements. No existing data models or implementations to draw from. Construct new data models, data dictionaries, database logical and physical design. This may require acquisition of DBMS. | Function points show potential for data costing; see [RUH91]. |
| Cost Maintenance | Continue database in its current form with upgrades within planned maintenance budget. | Current cost projections. |
| Reverse Engineering | Extracting data elements and models from existing files, application code, and documentation. | None available. |
| Forward Engineering | <p>There may be a change, for example, from a flat file data structure to a relational database. This would entail the acquisition of a relational DBMS, as well as the costs of manipulating the system data from the flat file structure to a relations-within-a-table structure. SSC estimates following data development costs:</p> <ul style="list-style-type: none">• \$40/data element for data standardization paperwork;• \$360/data element to modify code for standardized data;• \$600/data element to modify code for SQL access [SSC93]. | Function points show potential for data costing; see [RUH91] |
| Redocumentation | Construct and update any existing documentation including data models, data element descriptions, data dictionaries. | Estimates of labor costs. |

Infrastructure

The infrastructure area includes the hardware platform, the system software, and the network; and with a reengineering effort, it is very likely that there will be a change of processor, a change of operating system, as well as a change in the communications networks. These components most likely will be (or should be) provided from COTS acquisition, not through a new specialized development. One major consideration is that, most of the time, infrastructure components are COTS and need to be priced according to a per site basis. So the cost of a new site configuration must be multiplied by the number of sites for total infrastructure investment cost. Table 3-3 highlights the infrastructure reengineering cost estimation considerations and techniques for each activity type.

Table 3-3. Infrastructure Reengineering Cost Estimation

| Activity | Considerations | Cost Techniques |
|-----------------------|---|--|
| New Acquisition | NDI purchase or lease. Hardware costs: purchase/leasing, installation, maintenance. System software costs: operating system, device drivers, network software, etc. Yearly maintenance costs will vary, but generally run about 10% to 15% of acquisition costs. | Government purchase-contracts, GSA schedule |
| New Development | Although this option is not recommended, there may arise a situation where development or specialized configuration is necessary. | Specialized for item. |
| Continued Maintenance | Costs for existing platforms and system software. | Extrapolate current cost projections at minimum. |
| System Upgrades | Same as new acquisition; need to consider effect upon multiple sites and personnel time for installation. | Same as new acquisition. |

Example 3-9. Costed System Options

Costed System Options - A33

Baseline: Hardware maintenance costs will be \$200,000 per site per year: for 120 sites, the total annual cost is \$24 million. Software maintenance costs are \$40,000 for tools and \$460,000 for staffing, for a total of \$500,000 per year. Total annual system costs are \$24,500,000 per year.

Option A:

System Maintenance: Total Costs = \$24,500,00 per year.

Hardware and software maintenance costs are the same as Baseline.

System Investment: Total Costs = \$640,000.

-Software upgrade costs for Option A are estimated at \$640,000 (reverse engineering - \$90,000; forward engineering - \$500,000; tools - \$50,000).

Option B:

System Maintenance: Total Costs = \$2,900,000 per year.

-Software maintenance costs are the same as Baseline and Option A (\$500,000).

-Hardware maintenance costs per site are estimated at \$20,000 for total annual cost of \$2,400,000.

System Investment (Application, Data, Infrastructure): Total Costs = \$24,711,000.

Application: Total Costs = \$5,447,000.

-Reverse Engineering: 300,000 LOC; estimated 9 staff months is \$117,000.

-Forward Engineering: Estimated 200,000 LOC; at \$15 per LOC is \$3,000,000.

-New Development: Estimated 30,000 LOC - at \$60 per (reusable) LOC, \$1,800,000.

-Reengineering and development tools: \$50,000.

----- Reengineering Total Costs = \$4,967,000.

-Application COTS: Document preparation (Framemaker) is \$4,000 x 120 sites = \$480,000.

Data: Total Costs = \$3,922,000.

-Reverse Engineering: 200,000 LOC - Estimated 5,060 data elements (1,020 entities, 4,040 attributes) will require 4 staff months is \$52,000.

-Forward Engineering: Estimated final 1,500 data elements: 200 standardized at \$1,000 per data element is \$200,000; 1,300 non-standard at \$500 per data element is \$650,000.

-Reengineering and development tools: \$20,000.

----- Reengineering Total Costs = \$922,000.

-DBMS COTS: DBMS (Oracle) - \$25,000 x 120 sites = 3,000,000.

Infrastructure: Total Costs = \$15,342,000.

-Hardware COTS: 5 X-Terminals - \$40,000; SPARC-10 - \$70,000; Tape drives - \$10,000; other misc. hardware - \$5,000. total \$125,000.

-System software COTS: Operating System - \$1,2000; X-Windows - \$150; TCP/IP, utilities, etc. - \$1,500. Total = \$2850.

- Total infrastructure per site = \$127,850 x 120 sites = \$15,342,000.

3.3.4 Identify and Estimate Functional Costs (A34)

Description This activity consists of identifying and estimating functional costs for each system option. To assess the economic case in the functional process, a model should be used such as the Functional Economic Analysis Model (FEAM), which was specifically designed to assist DoD managers determine which business options would produce the best cost savings over a given period of time [IDA93].

- | | |
|--------------------------|---|
| <i>Inputs</i> | <ul style="list-style-type: none">• Risk-Assessed System Options• Benefit-Assessed System Options• Costed System Options |
| <i>Controls</i> | <ul style="list-style-type: none">• (Revised/Proposed) Functional Process Model• Reengineering Constraints & Objectives• Regulations, Policy, Standards, Guidelines |
| <i>Outputs</i> | <ul style="list-style-type: none">• Assessed System Options |
| <i>Mechanisms</i> | <ul style="list-style-type: none">• Functional Economic Analysis Model (FEAM)• Interviews with Functional Users• Project Documentation |

Considerations The current version of the FEAM (Version 3.0) is implemented as a PC-based tool and is supported by the Excel spreadsheet package. If the FEAM is used, then most of the calculations will be done automatically. The main responsibility of the user is to enter the cost estimates (including High and Low estimates) for each option into the appropriate categories. The tool will calculate the Net Present Value (NPV)² and Risk-Adjusted Discounted Cash Flow (RADCF)³ automatically. The user can also adjust factors such as discount rates as part of the tool's sensitivity analysis function. To have a complete business case, costs must be considered for at least six years.

The FEAM breaks down costs into two categories: Initiative and Operations. Initiative costs are those for one time or non-recurring activities, particularly those activities used to improve

² NPV is a capital budgeting method. It is the present value of benefits or cash inflows discounted at the project's cost of capital less the present value of the expected cost of the project.

³ RADCF is similar to NPV, the difference being that the discount rate (cost of capital) is adjusted up or down to account for risk.

the functional process. Operations costs are those for repeatable or recurring activities, specifically those to operate and maintain the existing functional process. In each category, costs are divided into seven cost elements (see Table 3-4).

Table 3-4. FEAM Cost Elements

| Initiative Category | Operations Category |
|--------------------------------|--------------------------------|
| A1. Civilian Labor | B1. Civilian Labor |
| A2. Military Labor | B2. Military Labor |
| A3. Equipment | B3. Equipment |
| A4. Material | B4. Material |
| A5. Facilities | B5. Facilities |
| A6. General and Administrative | B6. General and Administrative |
| A7. Other | B7. Other |

Assignment of Costs to FEAM Cost Elements

The assignment of costs to specific FEAM cost elements will require a degree of judgment by the user. The FEAM manual [IDA93] should be consulted for specific guidance on the assignment of costs to cost elements.

System development, maintenance, or operations costs incurred by government personnel, such as to build a piece of software, could be considered differently from COTS items regarding assignment of cost categories. In this case, the costs associated with system development, maintenance, or operations would be included in the "civilian labor," "military labor," or "other" cost elements. For example, it may be more appropriate to assign software development costs to a "labor" cost element under the Initiative Category; in other cases, it may be more appropriate to assign these costs to "other." Estimates for system support personnel should be applied to either "civilian labor" or "military labor" under the Operations Category.

Costs for hardware, software, and telecommunications equipment

(items that are purchased) should be included in the "equipment" or "material" cost elements.

Calculate NPV

The NPV is the cost savings, appropriately discounted, that will accrue over the life of the system. The FEAM calculates the NPV and accounts for the time value of money using discount rates provided by the Office of Management and Budget [OMB92]. The NPV is expressed as

$$\text{NPV} = \frac{\text{Baseline Cost} - \text{Options cost}}{(1 + \text{discount year})^{\text{year}}}$$

The FEAM User's Manual [IDA93] provides additional discussion and examples on the calculation of NPV.

Adjust for Risk

The FEAM tool then adjusts for risk by using the High and Low cost estimates for each option. The High estimate is the maximum that the estimate should not reasonably exceed. The Low estimate is the minimum that the estimate should not fall below. If risk is not a factor and costs are known with certainty, then the High and Low estimates are the same as the Total estimate. The risk-adjusted numbers reflect the difference of the High and Low estimates relative to the Total estimate. Thus a \$100 million investment with a High/Low difference of \$25 million is considered more risky than a \$200 million investment with the same High/Low difference. The result is the Expected RADCF, which is the NPV adjusted for risk. See [LUR93] for further discussion of cost risk analysis methods.

Example 3-10. Economic Analysis of System Options

Economic Analysis of System Options - A34

Note: In this example, only the costs in the selected categories are considered. all other costs are considered to be the same for each option. One person (i.e., one staff year) is estimated at \$115,000 per year. Discount rate for 6 years = 3.8%. The initiative costs are included in Year 1 only.

Baseline:

- A. Initiative = \$0
- B. Operations -> Total = \$93,500,000.
- B1. Civilian Labor -> (3 persons) \$345,000 x 120 sites = \$41,400,000.
- B2. Military Labor -> (2 persons) \$230,000 x 120 sites = \$27,600,000.
- B7. Other (Maintenance) -> (\$200,000 x 120 sites) + \$500,000 = \$24,500,000.

Option A:

- A. Initiative -> Total = \$640,000.
- A3. Equipment (Reengineering effort) = \$640,000
- B. Operations -> Total = \$70,500,000.
- B1. Civilian Labor -> (2 persons) \$230,000 x 120 sites = \$27,600,000.
- B2. Military Labor -> (2 persons) \$230,000 x 120 sites = \$27,600,000.
- B7. Other (Maintenance) -> (\$200,000 x 120 sites) + \$500,000 = \$24,500,000.

$$\begin{aligned} NPV &= (93.5 - 71.49) / (1+0.038)^1 + (93.5 - 70.5)/(1.038)^2 + (23)/(1.038)^3 + (23.0)/(1.038)^4 \\ &+ (23.0)/(1.038)^5 + (23.0)/(1.038)^6 = 21.532 + 21.355 + 20.56 + 19.812 + 19.08 + 18.388 \\ &= \$120,547,000. \end{aligned}$$

Option B:

- A. Initiative -> Total = \$24,711,000.
- A3. Equipment (\$25K min per item) = \$17,289,000.
- Application Reengineering (non COTS) = \$4,967,000
- Data Reengineering = \$922,000
- Hardware (SPARC-10) -> \$70,000 x 120 sites = \$8,400,000.
- COTS Software (Oracle) -> \$25,000 x 120 sites = \$3,000,000.
- A4. Material (under \$25K per item) -> \$7,422,000.
- Hardware -> \$55,000 x 120 sites = \$6,600,000.
- COTS Software -> \$6,850 x 120 sites = \$822,000
- B. Operations -> Total = \$41,400,000.
- B1. Civilian Labor -> (2 person) \$230,000 x 120 sites = \$27,600,000.
- B2. Military Labor -> (1 person) \$115,000 x 120 sites = \$13,800,000.
- B7. Other Maintenance -> (\$20,000 x 120 sites) + \$500,000 = \$2,900,000.

$$\begin{aligned} NPV &= (93.5-66.111) / (1+0.038) + (93.5 - 41.4)/(1.038)^2 + (52.1)/(1.038)^3 \\ &+ (52.1)/(1.038)^4 + (52.1)/(1.038)^5 + (52.1)/(1.038)^6 = 26.386 + 48.355 + 46.584 + 44.879 \\ &+ 43.238 + 41.65 = \$251,090,000. \end{aligned}$$

Example 3-11. Economic Analysis of System Options (Risk Adjusted)

Risk-Assessed System Options - A34 - (Risk Adjusted)

Adjusting For Risk

Option A:

A. Initiative -> Total = \$640,000.

-----> High = \$950,000.

-----> Low = \$500,000.

-- A3 Equipment (Reengineering effort) = \$640,000

B. Operations -> Total = \$70,500,000 (Certainty)

- B1 Civilian Labor -> (2 persons) \$230,000 x 120 sites = \$27,600,000.

- B2 Military Labor -> (2 persons) \$230,000 x 120 sites = \$27,600,000.

- B7 Other (Maintenance) -> (\$200,000 x 120 sites) + \$500,000 = \$24,500,000.

NPV = \$120,547,000 (without risk considered)

RADCF = \$120,200,000 (risk adjusted estimate)

Option B:

A. Initiative -> Total = \$24,711,000.

-----> High = \$28,000,000.

-----> Low = \$24,000,000.

-- A3 Equipment (\$25,000 minimum per item) = \$17,289,000.

----- Application Reengineering (non-COTS) = \$4,967,000.

----- Data Reengineering = \$922,000

----- Hardware (SPARC-10) -> \$70,000 x 120 sites = \$8,400,000.

----- COTS Software (Oracle) -> \$25,000 x 120 sites = \$3,000,000.

-- A4 Material (\$25,000 maximum per item) -> \$7,422,000.

----- Hardware -> \$55,000 x 120 sites = \$6,600,000.

----- COTS Software -> \$6,850 x 120 sites = \$822,000

B. Operations -> Total = \$41,400,000 (Certainty)

- B1 Civilian Labor -> (2 persons) \$230,000 x 120 sites = \$27,600,000.

- B2 Military Labor -> (1 person) \$115,000 x 120 sites = \$13,800,000.

- B7 Other (Maintenance) -> \$2,900,000

NPV = \$251,090,000 (without risk considered)

RADCF = \$248,100,000 (risk adjusted estimate)

3.4 SELECT BEST OPTION(S) (A4)

Objective The objective of this step is to select the most appropriate system option to pursue. Given assessments of the cost, benefit, and risk factors for each reengineering option, how should all this information be combined to produce a decision? This decision should be supported by some form of decision analysis and business-case rationale. Typically, a business case for investment in software reengineering must be defended in terms of reducing future operation and maintenance costs, increasing the benefit of services provided to the organization, and reducing the risk of not being able to deliver services the organization depends on. This section briefly describes several techniques that support making these decisions.

Activities This step consists of four activities (see Figure 3-6):

- *Determine Decision Criteria* to establish a basis for comparing system options.
- *Rank System Options* according to selected decision criteria.
- *Conduct Sensitivity Analysis* to assess the validity of the rankings.
- *Select Solution / Approach*, given the system rankings and sensitivity analysis.

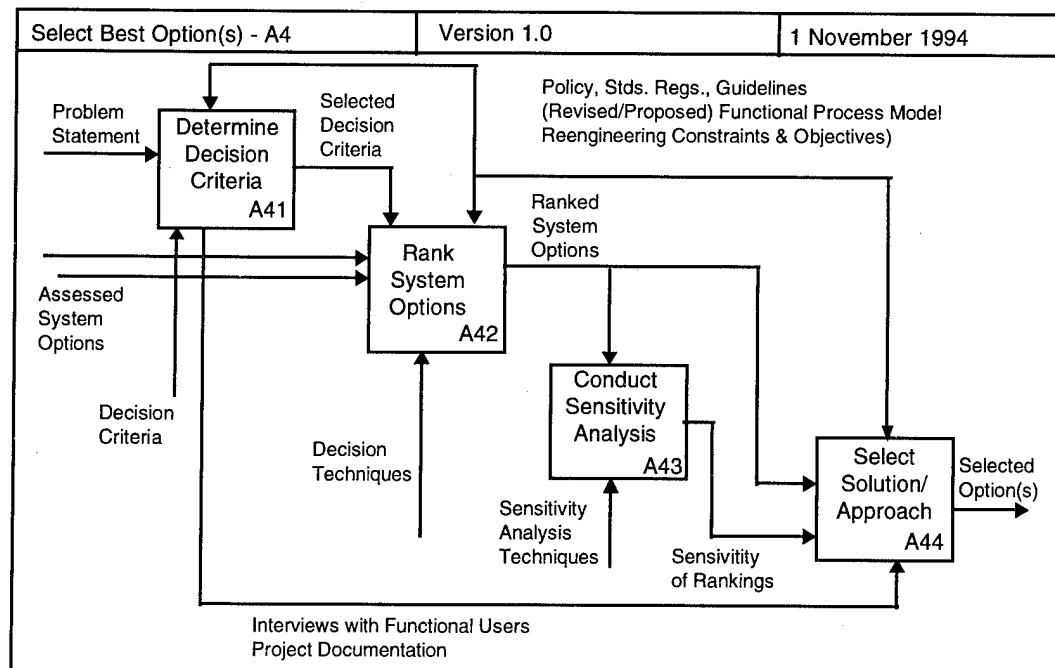


Figure 3-6. Select Best Option(s)

3.4.1 Determine Decision Criteria (A41)

Description This activity consists of determining the decision criteria to be used for ranking the assessed system options.

- | | |
|--------------------------|---|
| <i>Inputs</i> | • Problem Statement |
| <i>Controls</i> | • Reengineering Constraints & Objectives |
| | • (Revised/Proposed) Functional Process Model |
| | • Regulations, Policy, Standards, Guidelines |
| <i>Outputs</i> | • Selected Decision Criteria |
| <i>Mechanisms</i> | • Decision Criteria |
| | • Interviews with Functional Users |
| | • Project Documentation |

Considerations The highest NPV or RADCF should be the default criterion for choosing a system option. This criterion is specified by DoD Instruction 7041.3 and is used in the FEAM for evaluating economic decisions. If NPV is not required as the determining factor, then other decision criteria may be considered.

Decision Criteria

In this section, example decision criteria or objective functions are described that can be used to rank the available system options. Decisions are typically based on the highest ranking option that fits within the available budget. These functions can usually be expressed directly as mathematical formulas involving different sets of decision factors. In selecting decision criteria, we need to consider the overall mission objectives, lifetime of the system, and any short- and long-term economic needs (such as when is it necessary to realize a return on the investment).

- **Lowest Cost:** Cost has already been identified as a major decision factor. It could be used as the only factor to select between feasible options. This is the standard criterion for selecting between contract bids that are judged to meet the requirements as stated in a Request for Proposal (RFP). Note that, by itself, this criterion does not consider the accuracy of cost estimates or the potential for cost overruns.
- **Benefit to Cost Ratio:** Using cost alone as a selection criterion misses numerous opportunities to get more bang for the buck, when significant additional benefits can be achieved at some affordable additional cost. The difficult part of this criterion is coming up with accurate (or at least consistent)

measures of the benefits offered by different reengineering options.

- **Highest Return on Investment:** Return on investment for reengineering means the value of benefits achieved over time after subtracting the cost of reengineering. Benefits may be measured in terms of new functionality or reduced risk, as well as in operations and maintenance cost savings. The value of new functionality and reduced risks, however, must be translated into monetary terms by some means. Cost savings can be compared more directly by extrapolating expected costs over time with and without reengineering. Some reasonable time frame must be established as the point to measure expected return on investment.
- **Shortest Break-Even Return on investment:** This is another return on investment decision criterion, which is based on how long it takes for the reengineering improvements to pay for themselves. Using this metric rather than the highest-return metric could lead to a situation where a lower-return on investment option that pays back quickly is selected over one with a higher eventual return on investment but takes longer to become productive.
- **Best Available Technology:** Applying the best available technology in a reengineering effort may achieve the longest life-span solution. As a result, cost savings should be reflected in both operations and maintenance costs over the long term. Making the decision based on technology rather than cost, however, may avoid the uncertainties in estimating reengineering costs.
- **Hybrid Multi-Attribute Utility:** This approach combines any number of measures that reflect the value or utility of reengineering options. A composite utility formula is made up, giving weights to each utility factor. The advantage of this approach is that it makes the relative weights of each factor a conscious consideration and allows their adjustment to meet the task at hand. In practice, virtually all decisions are based on multiple factors.

Example 3-12. Selected Decision Criteria

| |
|--|
| Selected Decision Criteria - A41 |
| The selected decision criteria for this assessment will be highest NPV over a six-year period. |

3.4.2 Rank System Options (A42)

Description This activity consists of ranking system options with either informal or formal decision techniques.

- | | |
|--------------------------|---|
| <i>Inputs</i> | • Problem Statement |
| | • Assessed System Options |
| <i>Controls</i> | • Selected Decision Criteria |
| | • Reengineering Constraints & Objectives |
| | • (Revised/Proposed) Functional Process Model |
| | • Regulations, Policy, Standards, Guidelines |
| <i>Outputs</i> | • Ranked System Options |
| <i>Mechanisms</i> | • Decision Techniques |

Considerations Not all decision processes have to be complex; some decisions are easier, depending upon the inherent complexity of the problem. Other decision techniques are applied to problems that have more complicated decision criteria; these techniques break up the computation of complicated decision criteria into manageable steps. The following section discusses both informal and formal decision techniques.

Informal Decision Techniques

- **Intuitive Decisions:** If there are only a few options with very clear advantages and accurate cost estimates, and there is little uncertainty about future functional process needs, the decision may not require an elaborate decision process. Beware of completely subjective decisions, though. Even when more complicated decision methods are used, the ultimate decision must be presented in such a way that it makes intuitive sense.
- **Lists of Pros and Cons:** A popular technique for organizing and comparing decision factors is to tabulate the pros and cons for each option. People often find this helpful in weighing advantages and disadvantages. Assessment of the relative importance of the factors considered may still be somewhat subjective. The benefit of some new functionality, for example, might be judged to outweigh potential risks without an objective basis.

Formal Decision Techniques

The following subsections discuss example approaches that may be used to find the best solutions for more complicated decision problems. In general, these techniques are more quantitative and analytical, if somewhat less intuitive. An important pitfall to be aware of in these more analytical approaches is that, when two options end up with very nearly the same final rankings, the relative advantages or disadvantages between them may not be clearly distinguished. When this happens, you may be forced back into making an intuitive choice. The choices, however, will almost certainly be narrowed and the issues more sharply focused.

- **Delphi Technique:** The Delphi technique was developed at the Rand Corporation in the early 1950s as a method for forecasting events and estimating values that cannot be measured or derived analytically [LIN75]. The technique involves asking a group of experts (Rand suggests on the order of 20 to 60) to provide concrete answers to specific questions. Individual responses are kept anonymous. Composite summaries of the results are fed back to the experts for revised opinions. This process is repeated several times (usually four to five), until a consensus emerges or reasons why no consensus can be reached become clear. The anonymity of responses keeps strong personalities from dominating the results and allows experts to change their opinions without embarrassment or explanation. To avoid another obvious source of bias, the experts consulted should have no vested interest in the outcome. Experiments have shown that after each iteration, the consensus opinion (e.g., the mean or mode) usually moves toward the correct answer and the disparity among answers (e.g., the standard deviation) shrinks.

This approach can be applied to decision making by asking a group of experts to score or rank order the options in terms of the priorities set by the decision criteria. The more difficult part may be in finding enough experts who can quickly absorb all the information about the options, yet have no vested interest in the results to be produced.

- **Decision Trees:** Decision trees allow decisions to be made about parts of a solution approach, while deferring decisions about other parts. This may allow future hardware upgrade decisions to be postponed, for example, until the increased capacity is actually needed. Given expected reductions in the cost of hardware technology, delaying this part of the decision may be expected to result in lower costs for the hardware when it is needed, or allow

the purchase of more capacity for the same original cost.

Where complete solutions are needed, decision trees may help simplify the assessment of criteria by addressing them separately, rather than all at once. When conditional probabilities have to be introduced to represent decisions in tabular form, a decision tree can be used to simplify the analysis. Also, if consensus on a decision cannot be easily reached, decision trees may allow agreement to be reached on parts of a solution, which should help simplify and clarify the remaining issues.

- **Decision Matrices:** Decision matrices are tables that list the decision options along with all their associated factors. Expected outcomes and composite figures of merit are computed for each option and tallied in additional columns or in separate tables. The options are then ranked in terms of these composite measures. The highest ranking option represents the most strongly supported decision.
- **Analytical Hierarchy Process:** The Analytical Hierarchy Process approach was developed by Saaty in the early 1970s [SAA82]. It derives numerical weights for each of the decision factors and produces a composite priority figure for each option under consideration. A worked example of the use of this technique is given in Appendix A.

Decision factors are organized hierarchically, and at each level of the hierarchy, assessments are made of the relative importance of each factor compared with each of the other factors. Exact ratios are not critical. In fact, whole number comparisons are advocated.

These comparisons are used to derive a set of weights that represent the relative importance of each factor in the ultimate decision. Minor inconsistencies in ratings, which will occur with whole number comparisons, are easily tolerated. Significant inconsistencies can be detected, which allows offending ratings to be reconsidered.

All the weights computed are then used to derive an overall, composite, weighted priority that represents how well each option satisfies or supports the entire collection of decision factors. The highest priority option is the one that best satisfies the overall decision objectives as represented by the collection of pair-wise comparisons. The relative strengths and weaknesses of each option can be seen in the intermediate results.

Example 3-13. Ranked System Options

Ranked System Options - A42

Using NPV over six years, the options rank accordingly:

1. **Option B:** This option provides the highest NPV, \$251 million (nearly double of the next option) over six years.
2. **Option A:** This option provides the second highest NPV, \$120.5 million, a fairly high value given a relatively low investment cost (\$640,000).
3. **Baseline:** This option is ranked last: it provides for no cost savings.

Example 3-14. Ranked System Options - A Portfolio Example

Ranked System Options - A42 - A Portfolio Example

There may be the situation where we can choose more than one option. That is, our system options constitute a set of possible investment choices. The limiting factor is the amount of money to spend. This is often the situation in corporations that may have a preset capital budget. In this example, any combination of the following system options are possible to pursue, but the amount of investment money is limited.

Total Investment Budget = \$11 Million (M)

Options Ranked by NPV -----Total Invested

Options Pursued

- | | |
|--|---------|
| 1. Option C: NPV = \$25 M; Investment Cost = \$5 M; | \$5.0 M |
| 2. Option D: NPV = \$20 M; Investment Cost = \$3 M; | \$8.0 M |
| 3. Option E: NPV = \$15 M; Investment Cost = \$2.5 M; | \$0.5 M |

Options Not Pursued - past the \$11 M investment budget

- | | |
|--|---------|
| 4. Option F: NPV = \$12 M; Investment Cost = \$2 M; | \$2.5 M |
| 5. Option G: NPV = \$10 M; Investment Cost = \$2 M | \$4.5 M |

3.4.3 Conduct Sensitivity Analysis (A43)

Description This activity consists of assessing the sensitivity of the system rankings. Sensitivity analysis is a process in which decision factors are adjusted systematically to determine how sensitive the decision rankings are to estimated input factors.

- | | |
|--------------------------|-----------------------------------|
| <i>Inputs</i> | • none specified |
| <i>Controls</i> | • Ranked System Options |
| <i>Outputs</i> | • Sensitivity of Rankings |
| <i>Mechanisms</i> | • Sensitivity Analysis Techniques |

Considerations The sensitivity to potential estimate inaccuracies and uncertainty may be of significant importance in making decisions. For example, options may move up or down in their rankings when pessimistic estimates or forecasts are substituted for optimistic ones. If a decision is found to be strongly influenced by a particular factor, extra effort may be needed to ensure the estimate for this factor is as accurate as possible. If substantially improved estimates are not feasible, a less sensitive option may prove to be a better decision.

Non-trivial decisions almost always involve uncertainty. Most reengineering cost, benefit, and risk factors can only be estimated and a "best" decision must be made, often without knowing the accuracy of those estimates. Furthermore, the ideal approach to reengineering may not turn out so ideally if the estimates are incorrect or if actual circumstances take unexpected turns.

One sensitivity analysis technique is to tabulate optimistic, pessimistic, and "best" estimates for each decision factor. Starting with the best estimates for all factors, individual factors are changed, one at a time, to optimistic values and the rankings of options are reevaluated. This is repeated using pessimistic values. In each case, we want to see how the order of the highest-ranking options changes.

The next step in this analysis requires estimating the probabilities that the optimistic and pessimistic estimates will turn out to be accurate predictions. Tabulating the probable decision outcomes allows the option with the highest overall ranking and least sensitivity to uncertainty to be identified.

Sensitivity of decisions to potential budget changes should be

considered in light of today's military downsizing efforts. Reengineering approaches that can be scaled back without losing the utility of work already invested should be elevated in their ranking when budget uncertainty is factored into the evaluation criteria. Approaches that offer little utility until fully implemented should be ranked correspondingly lower.

Example 3-15. Sensitivity of Rankings

Sensitivity of Rankings - A43

The ability to realize a savings is affected primarily by the number of sites where the AMS is to be installed. The savings needs to be enough to justify the application software investment, which is approximately \$5 million. Investment hardware costs per site are offset by a corresponding drop in maintenance and personnel. If the AMS were to be deployed at fewer than 20 sites, then Option B may not be as cost effective. Otherwise, the rankings remained relatively the same. The number of installation sites was reviewed and it was determined that the minimum number of site installations was 98 sites, even with "worst case" downsizing scenarios.

3.4.4 Select Solution/Approach (A44)

Description This activity consists of selecting the solution or approach based upon the rankings and subsequent sensitivity analysis.

- | | |
|--------------------------|---|
| <i>Inputs</i> | <ul style="list-style-type: none">• Ranked System Options• Sensitivity of Rankings |
| <i>Controls</i> | <ul style="list-style-type: none">• Reengineering Constraints & Objectives• (Revised/Proposed) Functional Process Model• Regulations, Policy, Standards, Guidelines |
| <i>Outputs</i> | <ul style="list-style-type: none">• Selected Option(s) |
| <i>Mechanisms</i> | <ul style="list-style-type: none">• Interviews with Functional Users• Project Documentation |

Considerations Before selecting an option, it may be worthwhile to review or rethink the rankings (and any other parts of the analysis), particularly if the results are not expected.

- **Sanity Check Ranking of Options:** Rankings of options may show unexpected or non-intuitive results. Low rankings of intuitively popular options and high rankings of unpopular options should be challenged. Assessments of expected costs, benefits, and risks may need to be reconsidered.
- **Apply Budget Constraint:** Highly ranked options may have correspondingly high expected costs. Those outside the available budget will have to be rejected. There may be more than one possible option. See Section 3.4 for portfolio example.

Example 3-16. Selected Option(s)

| |
|----------------------------------|
| Selected Options(s) - A44 |
|----------------------------------|

| |
|---|
| <p>The selected system option is Option B, the full reengineering option with newly implemented application software, data bases, and hardware. This option is the most cost effective over six years while providing substantially increased functionality and extended system life.</p> |
|---|

4. SUMMARY OF SRAM METHOD

- Analyze Automated Information System and Environment (A1)
 - Conduct Problem Analysis (A11)
 - Identify Reengineering Objectives (A12)
 - Identify Reengineering Constraints (A13)
- Identify System Options (A2)
 - Identify System Reengineering Strategies (A21)
 - Assess Technology Alternatives (A22)
 - Develop System Configurations (A23)
- Estimate Costs, Risks, and Benefits (A3)
 - Identify and Estimate System Costs (A33)
 - Identify and Estimate Functional Costs (A34)
 - Select Best Option(s) (A4)
 - Determine Decision Criteria (A41)
 - Rank System Options (A42)
 - Conduct Sensitivity Analysis (A43)
 - Select Solution/Approach (A44)

APPENDIX A. ANALYTICAL HIERARCHY PROCESS EXAMPLE

This appendix describes the steps taken in solving a hypothetical decision problem using the Analytical Hierarchy Process (AHP) approach to decision making. In this hypothetical scenario there are two existing information systems already in operation, named Babbage and Hollerith, and one planned new system, named Turing. Babbage and Hollerith are aging systems and candidates for reengineering. The current budget, however, cannot support reengineering both **Babbage** and **Hollerith**, and developing the new **Turing**. So the problem is to figure out which combination of efforts would make the best use of available resources, maximizing the benefits and minimizing the risks.

The first step in the AHP approach is to create a hierarchy of decision factors. The top of the hierarchy represents the ultimate decision to be made. Each of the major contributing factors to be considered in the decision are placed at the second level of the hierarchy. Factors may be divided into additional, lower levels of contributing factors. At the bottom of the hierarchy, each possible option or possible outcome is listed.

Figure A-1 shows the hierarchy for our hypothetical situation. The decision about which combination of reengineering and development efforts to pursue is at the top. The second level shows the major decision factors, which for this example are cost, benefit, and risk. The third level shows the cost factor broken out into operation, maintenance, and investment costs. (Investment costs are intended to capture the costs of reengineering and development.) The bottom shows the available options which are to reengineer the Babbage system or to continue its operation and maintenance (O&M), to reengineer the Hollerith system or to continue its O&M, and to develop or defer developing the Turing system. The complete list of decision alternatives is shown in Table A-1. Note: A decision alternative constitutes any possible combination of available system options. For example, one alternative is to reengineer Babbage, continue O&M on Hollerith, and develop Turing.

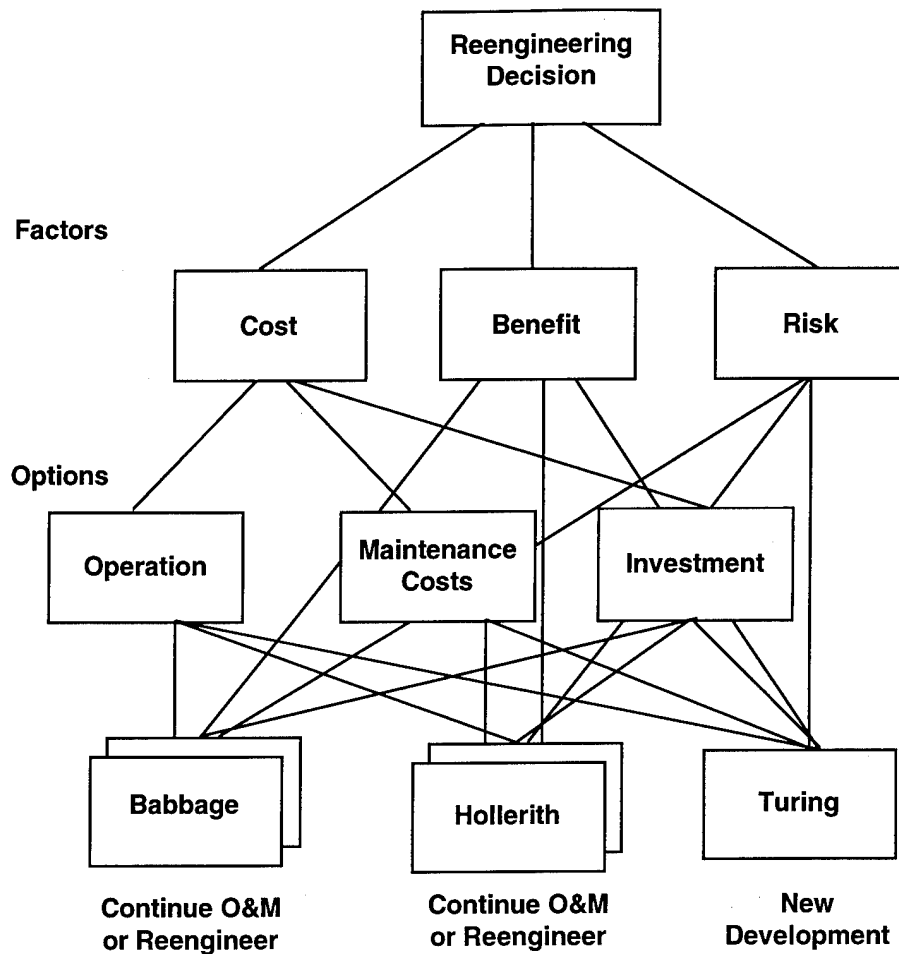


Figure A-1. Reengineering Decision Hierarchy

After completing the hierarchy, the next step is to rate the significance of contributing factors. Pairs of sibling factors are compared using the following rating scale:

- 1: No preference, the two factors are equally important.
- 3: Moderate preference of one factor over the other.
- 5: Strong preference of one factor over the other.
- 7: Very strong preference for one factor over the other.
- 9: Absolute preference for one factor over the other.

Table A-1. Decision Alternatives

| Alternative | Description |
|-------------|--|
| 1 | Continue Babbage & Hollerith O&M, no new development |
| 2 | Reengineer Babbage, continue Hollerith O&M, no new development |
| 3 | Reengineer Hollerith, continue Babbage O&M, no new development |
| 4 | Reengineer Babbage & Hollerith, no new development |
| 5 | Develop Turing, continue Babbage & Hollerith O&M |
| 6 | Reengineer Babbage, develop Turing, continue Hollerith O&M |
| 7 | Reengineer Hollerith, develop Turing, continue Babbage O&M |
| 8 | Reengineer Babbage & Hollerith, develop Turing |

For this example, overall cost is considered strongly more important than benefit (rating: 5), cost is considered moderately more important than risk (rating: 3), and risk is considered slightly more important than benefit (rating: 2). These ratings are then used to form the matrix shown in Table A-2.

Table A-2. Major Factors Comparison

| Factor | Cost | Benefit | Risk | Weight |
|--------------------------|------|---------|------|--------|
| Cost | 1 | 5 | 3 | 0.6483 |
| Benefit | 1/5 | 1 | 1/2 | 0.1220 |
| Risk | 1/3 | 2 | 1 | 0.2297 |
| Consistency Ratio | -- | -- | -- | 0.0032 |

The rows and columns of this matrix are identified by the factors compared. If the row factor is considered more significant than the column factor, the rating is placed in that position of the matrix. If the column factor is considered more significant than the row factor, the reciprocal of the rating ($1/\text{rating}$) is placed in that position of the matrix. Each factor is of equal importance to itself (rating: 1), so the major diagonal of the matrix contains all 1's. The eigenvalues of this matrix are then computed to derive a normalized weighting value for each factor. The weights add up to 1.0, and the ratios of pairwise weights are very

close to the ratings assigned.

A consistency check of the ratings that were assigned can also be computed. For this example, the consistency ratio is 0.0032. A ratio less than 0.01 indicates a consistent assignment of ratings.

The same analysis is applied to the cost factors. Operations costs are considered moderately more important to control than maintenance costs (rating: 3). Operations costs are considered very strongly more important to control than the investment costs (rating: 7). Maintenance is considered slightly more important to control than investment cost (rating: 2). These ratings are then used to form the matrix and the weighting values shown in Table A-3.

Table A-3. Cost Factors Comparison

| Factor | Oper. | Maint. | Invest. | Weight |
|--------------------------|-------|--------|---------|--------|
| Operations | 1 | 3 | 7 | 0.6817 |
| Maintenance | 1/3 | 1 | 2 | 0.2158 |
| Investment | 1/7 | 1/2 | 1 | 0.1025 |
| Consistency Ratio | -- | -- | -- | 0.0023 |

Table A-4 outlines one way to tabulate the estimates in each cost category. Continuing operation and maintenance of the existing systems with no reengineering has zero investment cost. Reengineering these systems will result in different (future) operation and maintenance costs. A decision not implement the new Turing system is assumed to incur no costs.

Table A-4. Cost Table

| Option | Operations | Maintenance | Investment |
|-------------------------|------------|-------------|------------|
| Babbage O&M | \$ _____ | \$ _____ | \$0 |
| Babbage Reengineering | \$ _____ | \$ _____ | \$ _____ |
| Hollerith O&M | \$ _____ | \$ _____ | \$0 |
| Hollerith Reengineering | \$ _____ | \$ _____ | \$ _____ |
| Turing Development | \$ _____ | \$ _____ | \$ _____ |

At the bottom of the hierarchy, estimates must be made for each pair of options, indicating their relative costs, benefits, and risks. Table A-5 lists the comparisons for each of the contributing cost factors. For example, the operating costs of continued use of the current Babbage system are estimated to be three times higher than the operating costs of the reengineered version of that system. Maintenance costs are also estimated to be three times higher for the current system than for the reengineered system. Investment costs are

not compared since continued operation and maintenance involves no investment. Reengineering the Babbage system, though, is estimated to cost twice as much as developing the Turing system (fourth row of Table A-5).

Table A-5. Pairwise Cost Comparisons

| Options Compared | Oper. | Maint. | Invest. |
|---|-------|--------|---------|
| Babbage O&M : Babbage Reengineering | 3 | 3 | N/A |
| Babbage O&M : Hollerith Reengineering | 2 | 4 | N/A |
| Babbage O&M : Turing Development | 3 | 5 | N/A |
| Babbage Reengineering : Turing Development | 1 | 1 | 2 |
| Hollerith O&M : Babbage O&M | 2 | 2 | N/A |
| Hollerith O&M : Babbage Reengineering | 5 | 7 | N/A |
| Hollerith O&M : Hollerith Reengineering | 4 | 8 | N/A |
| Hollerith O&M : Turing Development | 5 | 9 | N/A |
| Hollerith Reengineering : Babbage Reengineering | 1 | 1 | 3 |
| Hollerith Reengineering : Turing Development | 1 | 1 | 5 |

Table A-6 shows the matrix formed from the operations cost comparisons given in the first column of Table A-5. Table A-7 shows the corresponding maintenance costs matrix formed from the second column of Table A-5. Table A-8 shows the investment cost matrix, which includes entries only for those options that require investment spending.

Table A-6. Pairwise Cost Comparisons

| Option | Babbage O&M | Babbage Reeng. | Hollerith O&M | Hollerith Reeng. | Turing Dev. |
|-------------------------|-------------|----------------|---------------|------------------|-------------|
| Babbage O&M | 1 | 3 | 1/2 | 2 | 3 |
| Babbage Reengineering | 1/3 | 1 | 1/5 | 1 | 1 |
| Hollerith O&M | 2 | 5 | 1 | 4 | 5 |
| Hollerith Reengineering | 1/2 | 1 | 1/4 | 1 | 1 |
| Turing Development | 1/3 | 1 | 1/5 | 1 | 1 |

Table A-7. Maintenance Cost Matrix

| Option | Babbage O&M | Babbage Reeng. | Hollerith O&M | Hollerith Reeng. | Turing Dev. |
|-------------------------|-------------|----------------|---------------|------------------|-------------|
| Babbage O&M | 1 | 3 | 1/2 | 4 | 5 |
| Babbage Reengineering | 1/3 | 1 | 1/7 | 1 | 1 |
| Hollerith O&M | 2 | 7 | 1 | 8 | 9 |
| Hollerith Reengineering | 1/4 | 1 | 1/8 | 1 | 1 |
| Turing Development | 1/5 | 1 | 1/9 | 1 | 1 |

Table A-8. Investment Cost Matrix

| Option | Babbage Reeng. | Hollerith Reeng. | Babbage Dev. |
|-------------------------|----------------|------------------|--------------|
| Babbage Reengineering | 1 | 1/3 | 2 |
| Hollerith Reengineering | 3 | 1 | 5 |
| Turing Development | 1/2 | 1/5 | 1 |

Table A-9 lists the normalized weights for each option derived from each of these matrices. A composite weight for the overall cost of each option is then computed. This is done by multiplying the values from each row by the weights for each cost factor computed earlier (fourth column, Table A-3) and summing the results. These values, shown in the fourth column of Table A-9, indicate the relative overall weighted costs of each option.

Table A-9. Composite Cost Factors

| Option | Oper. | Maint. | Invest | Composite |
|--------------------------|--------|--------|--------|-----------|
| Babbage O&M | 0.2485 | 0.2651 | 0.0 | 0.2266 |
| Babbage Reengineering | 0.0929 | 0.0731 | 0.2297 | 0.1026 |
| Hollerith O&M | 0.4600 | 0.5323 | 0.0 | 0.4284 |
| Hollerith Reengineering | 0.1057 | 0.0668 | 0.6483 | 0.1530 |
| Turing Development | 0.0929 | 0.0627 | 0.1220 | 0.0894 |
| Consistency Ratio | 0.0044 | 0.0047 | 0.0032 | -- |

Table A-10 compares the estimated benefits of each option. There is a strong benefit (rating: 5) associated with reengineering the Babbage system compared with continuing to operate and maintain the current system. Reengineering the Hollerith system is considered equal in benefit (rating: 1) to developing the Turing system (next to last line). The matrix for this data is shown in Table A-11.

Table A-10. Pairwise Benefit Comparisons

| Options Compared | Relative Benefit |
|---|-------------------------|
| Babbage Reengineering : Babbage O&M | 5 |
| Babbage Reengineering : Hollerith O&M | 3 |
| Babbage Reengineering : Hollerith Reengineering | 2 |
| Babbage Reengineering : Turing Development | 2 |
| Hollerith O&M : Babbage O&M | 2 |
| Hollerith O&M : Turing Development | 1 |
| Hollerith Reengineering : Babbage O&M | 3 |
| Hollerith Reengineering : Hollerith O&M | 2 |
| Hollerith Reengineering : Turing Development | 1 |
| Turing Development : Babbage O&M | 3 |

Table A-11. Benefit Matrix

| Option | Babbage O&M | Babbage Reeng. | Hollerith O&M | Hollerith Reeng. | Turing Dev. |
|-------------------------|------------------------|-----------------------|--------------------------|-------------------------|--------------------|
| Babbage O&M | 3 | 1/5 | 1/2 | 1/3 | 1/3 |
| Babbage Reengineering | 5 | 1 | 3 | 2 | 2 |
| Hollerith O&M | 2 | 1/3 | 1 | 1/2 | 1 |
| Hollerith Reengineering | 3 | 1/2 | 2 | 1 | 1 |
| Turing Development | 3 | 1/2 | 1 | 1 | 1 |

Table A-12 compares the estimated risks of each option. There is a moderate risk (rating: 3) associated with continuing to operate and maintain the current Babbage system compared with reengineering it. All of the reengineering and development options are considered equal in risk (rating: 1) (rows 4, 5, and 10). The matrix for this data is shown in Table A-13

Table A-12. Pairwise Risk Comparisons

| Options Compared | Relative Benefit |
|---|------------------|
| Babbage O&M : Babbage Reengineering | 3 |
| Babbage O&M : Hollerith Reengineering | 4 |
| Babbage O&M : Turing Development | 2 |
| Babbage Reengineering : Hollerith Reengineering | 1 |
| Babbage Reengineering: Turing Development | 1 |
| Hollerith O&M : Babbage O&M | 4 |
| Hollerith O&M : Babbage Reengineering | 9 |
| Hollerith O&M : Hollerith Reengineering | 9 |
| Hollerith O&M : Turing Development | 7 |
| Hollerith Reengineering : Turing Development | 1 |

Table A-13. Risk Matrix

| Option | Babbage O&M | Babbage Reeng. | Hollerith O&M | Hollerith Reeng. | Turing Dev. |
|-------------------------|-------------|----------------|---------------|------------------|-------------|
| Babbage O&M | 1 | 3 | 1/4 | 4 | 2 |
| Babbage Reengineering | 1/3 | 1 | 1/9 | 1 | 1 |
| Hollerith O&M | 4 | 9 | 1 | 9 | 7 |
| Hollerith Reengineering | 1/4 | 1 | 1/9 | 1 | 1 |
| Turing Development | 1/2 | 1 | 1/7 | 1 | 1 |

The weights for each of the primary factors for each option are collected in Table A-14. The decision, though, is to be based not on the ratings of individual options, but on the best combination of options. These combinations were listed in Table A-1. To obtain the cost, benefit, and risk weights for each combination, add the cost, benefit, and risk weights for each option included. The option not to build the Turing system is assumed to have zero cost, benefit, and risk. Then normalize the columns so they sum to 1. The results are shown in Table A-15.

Table A-14. Cost, Benefit, & Risk Factors

| Option | Cost | Benefit | Risk |
|--------------------------|--------|---------|--------|
| Babbage O&M | 0.2266 | 0.0696 | 0.1872 |
| Babbage Reengineering | 0.1026 | 0.3386 | 0.0666 |
| Hollerith O&M | 0.4284 | 0.1394 | 0.6061 |
| Hollerith Reengineering | 0.1530 | 0.2150 | 0.0635 |
| Turing Development | 0.0894 | 0.1874 | 0.0766 |
| Consistency Ratio | -- | 0.0097 | 0.0116 |

The decision objective is to maximize benefit while minimizing cost and risk. Since the AHP technique attempts to maximize all weights, the cost and risk weights need to be transformed. The technique to minimize cost and risk is to maximize their reciprocals, $1/\text{cost}$ and $1/\text{risk}$. The columns of reciprocal values must be normalized so they sum to 1.

Table A-15. Decision Factors

| Alternative | Cost | Benefit | Risk |
|-------------|--------|---------|--------|
| 1 | 0.1638 | 0.0523 | 0.1983 |
| 2 | 0.1328 | 0.1320 | 0.1682 |
| 3 | 0.0949 | 0.0712 | 0.0627 |
| 4 | 0.0639 | 0.1509 | 0.0325 |
| 5 | 0.1861 | 0.0991 | 0.2175 |
| 6 | 0.1551 | 0.1788 | 0.1873 |
| 7 | 0.1172 | 0.1180 | 0.0818 |
| 8 | 0.0862 | 0.1977 | 0.0517 |

The composite weight for the overall priority of each alternative is computed by multiplying the $1/\text{cost}$, benefit, and $1/\text{risk}$ values by the weights for these factors computed earlier (fifth column, Table A-2) and summing the results. Table A-16 shows the normalized weights for the three major contributing factors and the final overall weights for each decision alternative.

Table A-16. Decision Priorities

| Alternative | 1/Cost | Benefit | 1/Risk | Priority |
|-------------|--------|---------|--------|----------|
| 1 | 0.0851 | 0.0523 | 0.0508 | 0.0732 |
| 2 | 0.1049 | 0.1320 | 0.0599 | 0.0979 |
| 3 | 0.1468 | 0.0712 | 0.1608 | 0.1408 |
| 4 | 0.2180 | 0.1509 | 0.3099 | 0.2310 |
| 5 | 0.0749 | 0.0991 | 0.0464 | 0.0713 |
| 6 | 0.0898 | 0.1788 | 0.0538 | 0.0924 |
| 7 | 0.1188 | 0.1180 | 0.1232 | 0.1197 |
| 8 | 0.1616 | 0.1977 | 0.1951 | 0.1737 |

The fourth alternative, reengineering both the Babbage and Hollerith systems, has the highest decision priority, 0.2310. Unfortunately, however, to make the example more realistic, we assume that the budget cannot support both of these activities. Table A-17 lists the alternatives, sorted by decision priority, along with their estimated this-year costs (normalized). This table shows that the second-priority alternative is even more costly than the first. The third-priority alternative, for this example, is assumed to be within budget and, therefore, becomes the best answer, being the highest-priority feasible solution.

Table A-17. Prioritized Decision List

| Priority | This-Year Cost | Alternative |
|----------|----------------|--|
| 0.2310 | 0.1689 | Reengineer Babbage & Hollerith |
| 0.1737 | 0.1962 | Reengineer Babbage & Hollerith, develop Turing |
| 0.1408 | 0.1403 | Reengineer Hollerith, continue Babbage O&M |
| 0.1197 | 0.1676 | Reengineer Hollerith, develop Turing, continue Babbage |
| 0.0979 | 0.0824 | Reengineer Babbage, continue Hollerith O&M |
| 0.0924 | 0.1097 | Reengineer Babbage, develop Turing, continue Hollerith |
| 0.0732 | 0.0583 | Continue Babbage & Hollerith O&M |
| 0.0713 | 0.0811 | Develop Turing, continue Babbage & Hollerith O&M |

LIST OF REFERENCES

- [BOE81] Boehm, B., *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [BOE89] Boehm, B., *IEEE Tutorial on Software Risk Management*, IEEE Computer Society Press, New York, 1989.
- [CAR93] Carr, M. J. et al., *Taxonomy-Based Risk Identification*, CMU/SEI-93-TR-6, Software Engineering Institute, Pittsburgh, PA, June 1993.
- [CHA88] Charette, R. N., *Software Engineering Risk Analysis and Management*, McGraw-Hill, New York, 1988.
- [CHI89] Chicken, John and Michael R. Haynes, *The Risk Ranking Technique in Decision Making*, Pergamon Press, 1989.
- [CHI90] Chikofsky, E. J. and Cross, J. H., "Reverse Engineering and Design Recovery: A Taxonomy," *IEEE Software*, January 1990, pp. 13-17.
- [CIM93a] Center for Information Management, *Automated Information Systems Software Reengineering Risks Taxonomy Report*, Defense Information Systems Agency, Joint Interoperability Engineering Organization, 1993.
- [CIM93b] Center for Information Management, *Information System Criteria for Applying Software Reengineering: Guidelines for Identifying Candidate Information Systems for Software Reengineering*, Defense Information Systems Agency, Joint Interoperability Engineering Organization, May 1993.
- [CIM93c] Center for Information Management, *Software Reengineering Project Planning Guide*, Version 1.0, Defense Information System Agency, Joint Interoperability Engineering Organization, October 1993.
- [CIM94] Center for Information Management, *Center for Information Management Software Systems Reengineering Process Model*, Version 2.0, Defense Information Systems

Agency, Joint Interoperability Engineering Organization, September 1994.

- [DOD94] *The DoD Enterprise Model*, OASD (C3I)/DDI, Arlington, VA 22202-4301, January 1994.
- [DOD92] *Functional Management Process for Implementing the Information Management Program of the Department of Defense*, DoD 8020.1-m (Draft), August 1992.
- [GUL93] Gullledge, T. and W. Hutzler, eds., *Analytical Methods in Software Engineering Economics*, Springer-Verlag, New York, 1993.
- [IDA86] *A Descriptive Evaluation of Automated Software Cost-Estimation Models*, Institute for Defense Analyses, Alexandria, VA, October 1982.
- [IDA93] *User's Manual for the Functional Economic Analysis Model (Version.3.0)*, Institute for Defense Analyses, Alexandria, VA, December 1993.
- [JLC93] Joint Logistics Commanders, *Reengineering Economics Handbook, Proceedings of First Software Reengineering Workshop - Santa Barbara I*, Joint Logistics Commanders Joint Policy Coordinating Group on Computer Resources Management, March 1993.
- [KIR92] Kirkpatrick, R. J. et al., *Software Development Risk Management: An SEI Appraisal*, Software Engineering Institute, Pittsburgh, PA, 1992.
- [LIN75] Linstone, Harold A. and Murray Turoff. *The Delphi Method: Techniques and Applications*, Addison-Wesley, 1975.
- [LUR93] Lurie, P. et al., *A Handbook of Cost Risk Analysis Methods*, IDA Paper P-2734, Institute for Defense Analyses, Alexandria, VA, April 1993.
- [MOA93] Moad, Jeff, "Does Reengineering Really Work?", *Datamation*, August 1, 1993, pp. 22-28.
- [MOR90] Morgan, M. Granger and M. Henrion, *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk*

and Policy Analysis, Cambridge University Press, 1990.

- [OMB92] Office of Management and Budget, Revised OMB Circular A-94, "Guidelines and Discount Rates for Benefit-Cost Analysis of Federal Programs," October 29, 1992.
- [RAI68] Raiffa, H., *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*, Addison-Wesley, 1968.
- [RUH91] Ruhl, M.K. and M.T. Gunn, *Software Reengineering: A Case Study and Lessons Learned*, National Institute of Standards and Technology Special Publication 500-193, September 1991.
- [SAA82] Saaty, Thomas L., *Decision Making for Leaders: The Analytical Hierarchy Process for Decisions in a Complex World*, Lifetime Learning Publications, Belmont, CA, 1982.
- [SSC93] *Standard Systems Center Base-Level System Modernization (BLSM) - A Strategy for the Future*, SSC/XON, Maxwell AFB-Gunter Annex, AL, 1 November 1993.
- [STS93a] Software Technology Support Center, *Reengineering Technology Report*, Hill Air Force Base, UT, August 1993.
- [STS93b] Software Technology Support Center, *Software Estimation Technology Report*, Hill Air Force Base, UT, March 1993.
- [WEL92] Wellman, Frank, *Software Costing*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

GLOSSARY

This glossary defines specific technical terms related to systems reengineering and the ICOMs (inputs, controls, outputs, mechanisms) of the SRAM and CIM Software Systems Reengineering Process.

Assessed System Options - ICOM in the SRAM. Candidate System Option(s) that have been evaluated according to cost, risk, benefits, and functional economic impact.

Automated Information System (AIS) - ICOM in the CIM Software Systems Reengineering Process Model. An AIS consists of any combination of computer hardware, computer software, telecommunications, information technology, personnel, data, documentation and other resources which collect, record, process, store, communication, retrieve, and display information. More than one system or parts of different systems may be input to the software reengineering activity. An AIS can include computer software only, computer hardware only, or any combination of the above [CIM94].

Automated Information System and Environment - ICOM in the SRAM. The existing combination of computer hardware, software, and documentation that support a specific activity within its setting of personnel, organization, and resources.

Available Reengineering Technology - ICOM in the CIM Software Systems Reengineering Process Model. Available Reengineering Technology identifies proposed methodologies and tools available for automating software reengineering. The Available Reengineering Technology constraints the Reengineering Project Plan, by affecting the Methodologies and Tools available for automating the software reengineering effort. It also affects the schedule and funding by the training necessary to use this technology and the productivity improvements in automating the software reengineering process. Repositories may exist that provide information on Available Reengineering Technology [CIM94].

Available Technologies - ICOM in the SRAM. Refers to any information or computing technologies (hardware, software, communications) that are readily accessible and implementable. This includes different methods, techniques, tools, and any equipment to support improvements in information management.

Baselined AIS Components - ICOM in the CIM Software Systems Reengineering Process Model. The selected information system components composed of the technical infrastructure, data, application software, and all associated documentation which will be used during reverse engineering [CIM94].

Benefit-Assessed System Options - ICOM in the SRAM. Candidate System Options that have been evaluated with regard to improvements in functionality, reliability, performance, usability, and compatibility, and maintainability.

Candidate Reuse Assets - ICOM in the CIM Software Systems Reengineering Process Model. Candidate Reuse Assets are potential reusable assets identified during the reengineering effort. Like Reusable Assets, the Candidate Reuse Assets are software work products, including source code, documentation, designs, test data, tools, lessons, learned, and specifications. These candidates are input to a reuse certification program for verification and validation as to potential usability in multiple software systems [CIM94].

Candidate Reuse Assets may describe repeatable processes such as reengineering strategies, maintenance processes, or new business practices.

Candidate System Options - ICOM in the SRAM. A set of system configuration that are defined in sufficient detail that analyses of cost, benefits, risk, and economic impact are possible.

Computing/Communications Infrastructure - ICOM in the CIM Software Systems Reengineering Process Model. A service utility that provides common shared computing and communications capabilities, including data base, common networks, electronic messaging, and computing platforms [CIM94].

Costed System Options - ICOM in the SRAM. Candidate System Options that have been assessed according to their expected cost in the areas of hardware and software acquisition, development, and maintenance.

Decision Criteria - ICOM in the SRAM. Factors by which Assessed System Options are compared. NPV is the default criterion; however, other types of decision criteria include lowest cost, benefit-to-cost ratio, and highest return on investment.

Decision Techniques - ICOM in the SRAM. Specific approaches for ranking Assessed System Options. These approaches can allow for the consideration of a variety of factors. Consisting of informal and formal techniques, decision techniques include Delphi, decision trees, decision matrices, and the Analytic Hierarchy Process.

Develop or Reengineer Systems - Activity within the CIM Information Management Process. This activity uses the inputs of Technical Requirements and Functional (Business) Requirements to produce the outputs of New or Reengineered System and Reverse Engineered Products.

DoD Enterprise Model - ICOM in the CIM Software Systems Reengineering Process Model. The DoD Enterprise Model is a representation of the activities and data of the Department of Defense (DoD) needed to accomplish the defense mission, from warfighting to acquisition and logistics support. This Model is the basis for defining, coordinating and integrating DoD missions and functions. It enables leaders and managers to better understand and direct their areas of responsibility, and to integrate functional process improvement initiatives within and across functional and organizational boundaries [CIM94].

Feasibility Analysis Results - ICOM in the CIM Software Systems Reengineering Process Model. The results from any study that may have been performed prior to the start of the reengineering project to scope the feasibility of reengineering should be used as input to the reengineering process. These results may identify and explore information necessary to perform the reengineering project. The results of this analysis should be input to the Software Systems Reengineering Process Model and the members of the Reengineering Process Team should participate in the performance of this analysis [CIM94].

The results of the analysis may include, but are not limited to, a cost/benefit analysis results, risk analysis and management, and a technical justification for the reengineering. The cost/benefit analysis determines the cost of performing the reengineering compared to the benefits expected from reengineering. The technical justification includes a description of how the reengineering project is justified based on the technical aspects of the effort [CIM94].

Forward Engineering - Within the context of reengineering, forward engineering is the software engineering activities that consume the products of reengineering activities, primarily reverse engineering, reuse, and new requirements to produce a target system [CIM94]. The goal is to create a software system via reengineering. This term primarily refers to the process of generating new software systems from reverse engineered designs. This term has evolved within reengineering to refer to those software engineering activities (traditionally performed during development) that are performed during or as a result of reengineering [CIM93b, p. 4].

Functional Process Improvement - An analysis of the existing functional (business) process to determine areas for improving efficiency and reducing costs. Also an activity within the CIM Information Management Process.

Functional (Business) Requirements - ICOM in the CIM Information Management Process. These requirements reflect desired improvements in the functional (business) process that have resulted from a functional process improvement analysis.

Functional Economic Analysis Model - ICOM in the SRAM. A modeling technique used to provide a comparative assessment of possible business decisions. The FEAM is also supported by an automated tool of the same name.

Functional Process Model (Revised/Proposed) - ICOM in the SRAM. A model of an organization's activities that depicts how an enterprise conducts its business. This model may be one that already exists, or is an improvement over previous models (revised), or is being considered as a replacement (proposed).

ICOM (Input, Control, Output, Mechanism) - Directed arc in the IDEF0 notation. Each ICOM represents an entity that is either required, used, adhered to, or produced by an IDEF0 activity.

Implement, Operate and Maintain - Activity within the CIM Information Management Process. Where the new information system (which supports a new functional process) is put into operation where it undergoes normal use and maintenance upgrades.

Interviews with Functional Users - ICOM in the SRAM. Interviews conducted with the participants of the functional process, including users of the existing system and any potential reengineered system.

Methodologies - ICOM in the CIM Software Systems Reengineering Process Model; ICOM in the SRAM. The system of principles, procedures, and practices applied to the project definition, development, operation, reengineering and support of a software system. Reengineering methodologies are subdivided into project definition, (and) reverse and forward engineering methodologies. These methodologies support various software engineering methodologies, which should be carefully investigated to ensure efficient technical integration into the sponsoring organization's existing software engineering environment [CIM94]. Examples of methodologies also include cost estimation, reverse engineering techniques, and decision making techniques.

New or Reengineered System - One or more products of the Develop or Reengineer Systems activity of the CIM Software Systems Reengineering Process Model [CIM94].

NPV (Net Present Value) - Capital budgeting method. It is the present value of benefits or cash inflows discounted at the project's cost of capital less the

present value of the expected cost of the project.

Operational Experience - Evidence obtained from the conduct of all Defense activities, including successes and failures in military operations, that represent the factual basis for assessing and improving the direction that guides Defense activities. [DOD94].

Problem Statement - ICOM in the SRAM. Description of the Automated Information System and Environment, including how the system fits into the current functional process and why improvement or possible reengineering may be necessary.

Project Documentation - ICOM in the SRAM. Any documentation associated with the Automated Information System and Environment, including requirements, design, and test specifications; user and system manuals; project management plans and schedules; and reviews and project reports.

Project Team - ICOM in the CIM Software Systems Reengineering Process Model. The personnel who will perform the reengineering effort form a team. The members of this team may include, but not limited to experts, in the following areas: software/systems engineering, technical infrastructure, function/mission of the system domain, users of the application software, and reengineering technology. Specifically, the Project Team should involve the functional customer as much as possible throughout the reengineering effort [CIM94].

RADCF (Risk-Adjusted Discounted Cash Flow) - Is similar to NPV, the difference being that the discount rate (cost of capital) is adjusted up or down depending on the perceived riskiness of the project.

Redocumentation - Redocumentation produces supplementary information that provides understanding of the existing system and its components. This activity is usually performed to assist in maintenance of existing systems. This activity does not alter the existing software system representation, nor does it generate any new representation to replace any part of the existing representation [CIM93b, p. 4].

Redocumentation is often performed as part of reverse engineering to produce interim documentation that is used to generate or is converted to reverse engineered products, (e.g., business rules, data models, and process models).

Reengineering Constraints - ICOM in the SRAM. Specific limitations on potential system solutions. These limitations may fall in different areas such as budgetary, technical, organizational, or political.

Reengineering Objectives - ICOM in the SRAM. Specific improvements that the

newly reengineered system should implement.

Reengineering Options - ICOM in the SRAM. General system reengineering strategies that are constructed to provide the basis for further system configuration. These options are not detailed but represent broad differences in approach such as simple maintenance upgrades vs. full-scale reengineering.

Reengineering Project Plan - ICOM in the CIM Software Systems Reengineering Process Model. The Reengineering Project Plan documents the Objectives, identifies the Baseline AIS Components, the Project Resources, and Project Strategy. This plan includes refined analysis results, risk analysis/management information, and a formalization of the Business Requirements for the Reengineering System. The requirements available in the Baseline AIS Components are confirmed through the reverse engineering process and those to be implemented during forward engineering are identified as part of the Analysis Deliverables [CIM94].

Reengineered System - ICOM in the CIM Software Systems Reengineering Process Model. The reengineered system is generated from the reengineering activities described within this model. It consists of software, data, technical infrastructure, test results, and all associated documentation [CIM94].

Regulations, Policy, Standards, Guidelines - ICOM in the CIM Software Systems Reengineering Process Model and the SRAM. Documents containing the principle rules designed for governing and influencing decisions and actions during software engineering activities [CIM94]. There are many such documents, two of which are the Automated Information Systems Software Reengineering Risks Taxonomy [CIM93a] and the Information Systems Criteria for Applying Software Reengineering [CIM993b].

Repositories - ICOM in CIM Software Systems Reengineering Process Model. A mechanism for storing and retrieving information or reusable assets. Examples of repositories include the Defense Software Repository System (DSRS), DoD Data Repository System (DDRS), Integrated Computer-Aided Software Engineering (I-CASE), and DoD IDEF Repositories. The DDRS and the DSRS are managed by the CIM Data Administration Program Office and the Reuse Program Office respectively. The DoD IDEF Repository is managed by the CIM Center for Expertise in Functional Process Improvement (FPI). Repository-based technology may also be used to store and retrieve information generated during the reengineering project, including Reverse Engineered Products and the Reengineered Systems components [CIM94].

Resource Limitations - ICOM in CIM Software Systems Reengineering Process Model and SRAM. Estimated limitations on available resources, including

manpower, funding, scheduling deadlines, computer resources, and skill levels for performing the reengineering [CIM94].

Restructuring - The transformation of a software system from one representation form to another, while preserving the external behavior both (functionally and semantically) [CHI90]. The goal is to improve the existing structure without altering the functionality [CIM93b, p. 4].

Reverse Engineering - The process of examining an information system by analyzing its documentation, application software, and data structures within the environment in which the information system operates [CIM93b, p. 3]. This analysis is performed to (1) identify the system's components and their interrelationships, and (2) create representations of the system in another form or at a higher level of abstraction [CHI90]. The goal is to understand the existing software system (functions, performance, or implementation). Extracted information is represented in a format which can be integrated into the life cycle for development of a software system [CIM93b, p. 3].

Reverse Engineered Products - Products resulting from the reverse engineering effort which are used in the forward engineering process. These products include, but are not limited to, the business rules, refined feasibility analysis results, updated risk analysis, design model, system specification, functional requirements, metric data, data models, process models, and design decisions. Reverse engineered products reveal the business requirements fulfilled by the existing AIS [CIM94].

Reverse Engineering Techniques - ICOM in SRAM. Methods, either automated or manual, that allow the extraction of design or requirements information from existing system implementations. See Reverse Engineering.

Risk-Assessed System Options - ICOM in SRAM. Candidate System Options that have been evaluated with regard to risk in the areas of planning, process, personnel, product, and technology.

Selected Option(s) - ICOM in SRAM. Final output of the entire SRAM that represents the final system solution(s) that has been chosen for implementation. This solution(s) would be then used by the CIM Software Systems Reengineering Process Model.

Sensitivity Analysis Techniques - ICOM in SRAM. Techniques by which to assess the potential uncertainties or inaccuracies in decision factor values. Varying the value of these factors allows the determination of which ones have greater or lesser effect upon the final decision.

Sensitivity of Rankings - ICOM in SRAM. Assessment of ranked system options.

This assessment reflects the probabilities that a particular system option will occur.

Software Reengineering - Is composed of activities supporting the development and maintenance of automated information systems based on the examination and utilization of existing software system resources [CIM94]. The process encompasses a combination of other processes such as reverse engineering, restructuring, forward engineering, redocumentation, and translation. The goal is to improve the software system (functionality, performance, or implementation). Additional functionality is often incorporated into the system during this process. [CIM93b, p. 3].

Software Reuse - The application of existing software work products, including source code, documentation, designs, test data, tools, and specifications, in a software development effort other than the one for which each was originally developed. The goal is to facilitate the return on investment (ROI); improve software quality and reliability; shorten system development and maintenance times; increase productivity and minimize software-related risks. Software reuse should be employed during reengineering and reengineering should be applied to identify candidate reusable assets [CIM93b, p. 4].

SRAM (System Reengineering Assessment Method) - Method to assess the comparative costs, risks, and benefits of reengineering information systems. This method, as described in this document, consists of four major activities: (1) analyze automated information system and environment, (2) identify system options, (3) estimate costs, risks, and benefits, and (4) select best option(s).

Technical Architectures - ICOM in CIM Software Systems Reengineering Process Model and SRAM. Representation of the structure of technical infrastructure components, including computer platforms, support software, and communications, their relationships and interactions [CIM94].

Technical and Economic Analysis - The dual activities of technical and economic analyses of potential improvements to information system support are conducted to determine which system option will best support the new functional process. Also an activity within the CIM Information Management Process.

Technical Improvement Opportunities - May result from technology changes identified in the Perform Technical & Economic Analysis activity of the CIM Software Systems Reengineering Process Model [CIM94].

Technical Requirements - Technical Changes identified in the Perform Technical & Economic Analysis Activity of the CIM Software Systems Reengineering

Process Model which are designated for the New or Reengineered Systems [CIM94].

Techniques, Tools, and Models - ICOM in SRAM. Techniques, tools, and models for software reengineering, cost and economic analysis, and decision making.

Tools - ICOM in CIM Software Systems Reengineering Process Model and SRAM. Automated and manual implements used to improve productivity in performing or accomplishing the activities. These tools should integrate into the sponsoring organization's software engineering environment. Several organizations currently support tool evaluation and should be contacted to support the selection of tools appropriate for the individual needs of the reengineering project [CIM94]. Transition Plan and Training -Provides a plan to Implement, Operate, and Maintain the New or Reengineered System; and to provide adequate training for this transition to succeed [CIM94].

Transition to Operation and Maintenance - This activity uses the inputs of New or Reengineered System and Operational Experience to produce the output, Transition Plan & Training.

Translation - Transformation of source code from one language to another, or from one version of a language to another version of the same language. The goal is to improve the linguistic implementation of the software. This process is most successful when the two languages are similar or have a defined mapping between syntax [CIM93b, p. 4].

LIST OF ACRONYMS

| | |
|------|--|
| 4GL | Fourth Generation Programming Language |
| AFB | Air Force Base |
| AHP | Analytic Hierarchy Process |
| AIS | Automated Information System |
| AMS | Aircraft Maintenance System |
| BLSM | Base-Level System Modernization |
| CASE | Computer-Aided Software Engineering |
| CDA | Central Design Activity |
| CIM | Center for Information Management |
| COTS | Commercial off the shelf |
| CPU | Central Processing Unit |
| CSCI | Computer Software Configuration Item |
| DBMS | Database Management System |
| DDRS | DoD Data Repository System |
| DLA | Defense Logistics Agency |
| DoD | Department of Defense |
| DSRS | Defense Software Repository System |
| FEA | Functional Economic Analysis |
| FEAM | Functional Economic Analysis Model |
| FPI | Functional Process Improvement |
| GB | Gigabyte |

| | |
|--------|---|
| GSA | Government Services Administration |
| GUI | Graphical User Interface |
| I-CASE | Integrated Computer-Aided Software Engineering |
| ICOM | Input, Control, Output, Mechanism |
| IDA | Institute for Defense Analyses |
| I/O | Input/Output |
| IS | Information System |
| LOC | Lines of Code |
| MB | Megabyte |
| NDI | Non-Developmental Item |
| NPV | Net Present Value |
| O&M | Operation and Maintenance |
| OOT | Object-Oriented Technology |
| POSIX | Portable Operating System Interface for Computer Environments |
| RADCF | Risk-Adjusted Discounted Cash Flow |
| RAM | Random Access Memory |
| RDBMS | Relational Database Management System |
| RFP | Request for Proposal |
| ROI | Return on Investment |
| SRAM | System Reengineering Assessment Method |
| SSC | (USAF) Standard Systems Center |
| TAFIM | Technical Architecture Framework for Information Management |
| TCP/IP | Transmission Control Protocol/Internet Protocol |

| | |
|------|---------------------------|
| TRM | Technical Reference Model |
| USAF | United States Air Force |